RENCANA PEMBELAJARAN SEMESTER

(RPS)

MATA KULIAH: Basis Data Lanjut

PROGRAM STUDI S1 SISTEM INFORMASI

FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS

2017

Mata kuliah : Basis Data Lanjut
Kode Mata kuliah : TIK
S K S : 3 SKS
Prodi Konsentrasi : Sistem Informasi
Semester : 5
Kode Dosen : 196404091995121001

A. Deskripsi Mata kuliah

Mata kuliah ini membahas tentang basis data spasial PostGreSQL/PostGIS dan pemakaian perintah-perintah operasi spasial PostGIS di dalam mengembangkan satu sistem informasi spasial.

B. Capaian Pembelajaran (Kompetensi yang diharapkan)

Capaian pembelajaran mata kuliah :

1. Mahasiswa mampu memahami basis data PostGreSQL /PostGIS untuk mengembangkan sistem informasi spasial
2. Mahasiswa mampu memahami sistem informasi spasial yang ada.
3. Mahasiswa mampu mengembangkan sistem informasi spasial yang ada.

C. Capaian Pembelajaran dan Materi Pembahasan setiap pertemuan

Pertemuan	Kemampuan akhir yg diharapkan	Materi Pembelajaran (Bahan Kajian)
1, 2,3,4	Mhs mampu memahami basis data PostGreSQL /PostGIS untuk mengembangkan sistem informasi spasial	Database PostgreSQL dan PostGIS
5, 6, 7	Mhs mampu memahami sistem informasi spasial yang ada.	Aplikasi GIS untuk masjid, kuliner, tanah dan penduduk
8	**Evaluasi Tengah Semester**	
9, 10, 11, 12, 13, 14, 15	Mhs mampu mengembangkan sistem informasi spasial yang ada	Pengembangan Aplikasi GIS untuk masjid, kuliner, tanah dan penduduk
16	**Evaluasi Akhir Semester**	

D. Kemampuan Akhir Hard skills dan Softskill melalui Mata Kuliah

Kemampuan Hardskills	**Kemampuan** Softskills:
1. memahami basis data PostGreSQL /PostGIS untuk mengembangkan sistem informasi spasial 2. memahami sistem informasi spasial yang ada. 3. mengembangkan sistem informasi spasial yang ada	1. Bekerjasama dalam tim 2. Komunikasi secara efektif

E. Strategi Perkuliahan

Perkuliahan akan diberikan dalam 14 kali pertemuan, termasuk di dalamnya ujian tengah semester (UTS) dan ujian akhir semester (UAS) berupa penyerahan laporan tugas besar. Kegiatan tatap muka diisi dengan diskusi, presentasi dan tugas aplikasi. Kegiatan mandiri diisi dengan penelaahan/pengkajian teori pada buku/literatur yang dianjurkan. Kegiatan mandiri ini **wajib** dilakukan oleh setiap peserta di luar kegiatan tatap muka. Kegiatan terstruktur diisi dengan tugas-tugas pengayaan dan pendalaman. Kehadiran mahasiswa dalam perkuliahan minimal **75 persen** dari jumlah pertemuan yang diselenggarakan.

F. Evaluasi Perkuliahan

Keberhasilan dalam mengikuti mata kuliah ini didasarkan atas penilaian terhadap hasil-hasil pekerjaan mahasiswa, yang memenuhi persyaratan kehadiran minimal 75%, ujian tengah semester (bobot 1), dan ujian akhir semester (bobot 2). Skor akhir akan diolah dengan menggunakan Acuan Patokan, dan dikonversi ke dalam nilai A B C D E.

G. Sumber rujukan

1. PostGreSQL
2. PostGIS.
3. Aplikasi GIS untuk mesjid, kuliner, penduduk dan bangunan (Hasil riset)

H. Penanggung Jawab Matakuliah

1. Pengampu mata kulaiah

Prof. Dr. Surya Afnarius

RENCANA PEMBELAJARAN SEMESTER (RPS)
PROGRAM STUDI : Sistem Informasi
FAKULTAS /PPs: Teknologi Informasi
UNIVERSITAS ANDALAS

MATA KULIAH		**KODE**	**Rumpun MK**	**BOBOT (sks)**	**SEMESTER**	**Tgl Penyusunan**
Basis data lanjut		**TSI**	**Matakuliah Inti Keilmuan**	**3**	**5**	**2-8-2017**
OTORISASI		**Dosen Pengembang RPS**	**Koordinator Rumpun MK**		**Ka Program Studi**	
Capaian Pembelajaran (CP)	**CP Program Studi**					
	S9	Menunjukkan sikap bertanggungjawab atas pekerjaan di bidang keahliannya secara mandiri				
	P1	Menguasai konsep teoritis bidang pengetahuan Sistem Informasi secara umum dan konsep teoritis bagian khusus dalam bidang pengetahuan tersebut secara mendalam, serta mampu memformulasikan penyelesaian masalah prosedural.				
	P2	Menguasai konsep teoritis yang mengkaji, menerapkan dan mengembangkan serta mampu memformulasikan dan mampu mengambil keputusan yang tepat dalam penyelesaian masalah.				
	KU2	Mampu menunjukkan kinerja mandiri, bermutu, dan terukur;				
	KU10	Mampu melakukan analisis & desain dengan menggunakan kaidah rekayasa software dan hardware serta algoritma dengan cara menggunakan tools dan dapat menunjukkan hasil dan kondisi yang maksimal untuk aplikasi bisnis.				
	KU11	Memiliki kemampuan untuk menjadi tenaga profesional untuk pengolahan basis data, rekayasa SI, jaringan komputer, komputer grafis, dan aplikasi multimedia serta memiliki kemampuan menulis laporan penelitian dengan baik serta mengelola proyek Sistem Informasi, mempresentasikan karya tersebut.				
	KK3	Mampu memelihara dan mengembangkan sistem database (MsSQL/Oracle/PostGreSQL) yang digunakan oleh perusahaan untuk menyimpan, menganalisis, dan mengambil data serta bertanggung jawab terhadap kinerja, integritas dan keamanan database.				
	CP Mata Kuliah					
	1	Mahasiswa mampu memahami basis data PostGreSQL /PostGIS untuk mengembangkan sistem informasi spasial (KU11, KK3)				

Catatan :
S : Sikap
P : Pengetahuan
KU : Keterampilan Umum
KK : Keterampilan Khusus

	2	Mahasiswa mampu memahami sistem informasi spasial yang ada (P1, P2, KU2, KU10)
	3	Mahasiswa mampu mengembangkan sistem informasi spasial yang ada (S9, KU2,KU10, KU11, KK3)
	4	Mahasiswa memiliki kemampuan softskill dalam pembelajaran berupa: - Mampu berkomunikasi lisan dengan baik - Mampu bekerja sama dalam kelompok - Mampu mengelola / leadership dalam kelompok.
Deskripsi Singkat Mata Kuliah	Mata kuliah ini membahas tentang PostGreSQL/PostGIS sebagai database spasial dan pengembangan aplikasi GIS yang ada seperti aplikasi GIS untuk masjid, kuliner, tanah dan penduduk.	
Materi Pembelajaran/ Pokok Bahasan	**Materi Ajar :** PostGreSQL PostGIS Aplikasi GIS untuk masjid, kuliner, tanah dan penduduk	
Pustaka	**Utama :** 1. PostGreSQL 2. PostGIS. 3. Aplikasi GIS untuk mesjid, kuliner, penduduk dan bangunan (Hasil riset) **Pendukung :** -	
Media Pembelajaran	**Perangkat Lunak :**	**Perangkat keras :**
	Aplikasi GIS untuk mesjid, kuliner, penduduk dan bangunan	Infocus
Team Teaching	1. Prof. Surya Afnarius, PhD	
Assessment		
Matakuliah Syarat	-	

Mg Ke-	**Kemampuan akhir yg diharapkan**	**Bahan Kajian (Materi Ajar) Dan Referensi**	**Metode Pembelajaran dan Alokasi Waktu**	**Pengalaman Belajar Mahasiswa**	**Kreteria (Indikator) Penilaian**	**Bobot Penilan (%)**
(1)	(2)	(3)	(4)	(5)	(6)	(7)
1, 2,3,4	memahami basis data PostGreSQL /PostGIS untuk mengembangkan sistem informasi spasial (KU11, KK3)	PostGreSQl/PostGIS	Kuliah dan diskusi, (TM;4x(3x50")	Mahasiswa membentuk kelompok dan mencari informasi dari berbagai sumber (terutama Internet).		
5,6,7	memahami sistem informasi spasial yang ada (P1, P2, KU2, KU10)	Aplikasi GIS untuk mesjid, kuliner, tanah dan penduduk	Kuliah dan diskusi, (TM;3x(3x50")	Mahasiswa mengkaji jalanya aplikasi dan menambahkan fungsional tambahan.		
8	UTS/ Tugas Besar				Adanya fungsional-fungsional tambahan untuk aplikasi GIS mesjid, kuliner, tanah atau penduduk.	20
9,10,11 ,12,13, 14,15	mengembangkan sistem informasi spasial yang ada (S9, KU2,KU10, KU11, KK3)	Aplikasi GIS untuk mesjid, kuliner, tanah dan penduduk	Kuliah, diskusi, presentasi (TM;7x(3x50")	Mahasiswa mengimplementasikan fungsional-fungsional tambahan aplikasi GIS mesjid, kuliner, tanah atau penduduk..		
16	UAS/ Tugas Besar				Fungsional-fungsional tambahan pada aplikasi GIS mesjid, kuliner, tanah atau penduduk telah diimplementasikan	**80**

	PROGRAM STUDI : Sistem Informasi **FAKULTAS /PPs: Teknologi Informasi** **UNIVERSITAS ANDALAS**				
RENCANA TUGAS MAHASISWA					
MATA KULIAH	Basis data lanjut				
KODE	TSI	**sks**	3	**SEMESTER**	5
DOSEN PENGAMPU	Prof. Surya Afnarius, PhD				
BENTUK TUGAS					
Final Project					
JUDUL TUGAS					
Tugas: Final Project: Mengembangkan aplikasi GIS yang ada dan mempresentasikan.					
SUB CAPAIAN PEMBELAJARAN MATA KULIAH					
- Mahasiswa mampu mengembangkan sistem informasi spasial yang ada					
DISKRIPSI TUGAS					
Mengembangkan sistem informasi spasial yang ada					
METODE PENGERJAAN TUGAS					
1. Memilih satu SI spasial yang ada; 2. Mempelajari cara kerja SI spasial tsb; 3. Menghasilkan luaran SI; 4. Menyusun laporan; 5. Menyusun bahan & slide presentasi laporan; 6. Presentasi laporan di klas.					
BENTUK DAN FORMAT LUARAN					
a. Obyek Garapan: satu SI spasial yang ada **b. Bentuk Luaran:** 1. Laporan ditulis dengan MS Word dengan sistematika dan format sesuai dengan standar panduan penulisan, dikumpulkan dengan format ekstensi (*.rtf), dengan sistimatika nama file: (**Tugas-laporan-no nrpmhs-nama depan mhs.rtf**); 2. Slide Presentasi PowerPoint, terdiri dari : Text, grafik, tabel, gambar, animasi ataupun video clips, minimum 10 slide. Dikumpulkan dlm bentuk *softcopy*format ekstensi (*.ppt), dengan sistimatikan nama file: (**Tugas-Slide-no nrpmhs-nama depan mhs.ppt**);					
INDIKATOR, KRETERIA DAN BOBOT PENILAIAN					
a. Laporan (50%) Fungsional-fungsional yang dikembangkan dan Kesesuaian data di dalam database dengan luaran yang dihasilkan. **b. Penyusunan Slide Presentasi (bobot 20%)** Jelas dan konsisten, Sedehana & inovative, menampilkan gambar & blok sistem, tulisan menggunakan font yang mudah dibaca, jika diperlukan didukung dengan gambar dan vedio clip yang relevant. **c. Presentasi (bobot 30%)** Bahasa komunikatif, penguasaan materi, penguasaan audiensi, pengendalian waktu (15 menit presentasi + 5 menit diskusi), kejelasan & ketajaman paparan, penguasaan media presentasi.					
JADWAL PELAKSANAAN					
Mengembangkan SI spasial yang ada	Minggu I s/d Minggu 11				
Menyusun laporan	Minggu 12 s/d Minggu 13				
Presentasi laporan	Minggu 14 s/d 15				
Pengumuman nilai	Dua minggu setelah UAS				
LAIN-LAIN					
Bobot penilaian tugas ini adalah 100% dari penilaian mata kuliah ini; Tugas dikerjakan dan dipresentasikan secara kelompok;					
DAFTAR RUJUKAN					
- Aplikasi GIS untuk mesjid, kuliner, penduduk dan bangunan (Hasil riset)					

Minggu 1, 2, 3 dan 4

PostGreSQl/PostGIS

Bahan yang perlu dibaca sebagai persiapan diskusi dapat dilihat pada situs ini:

http://www.postgresqltutorial.com/

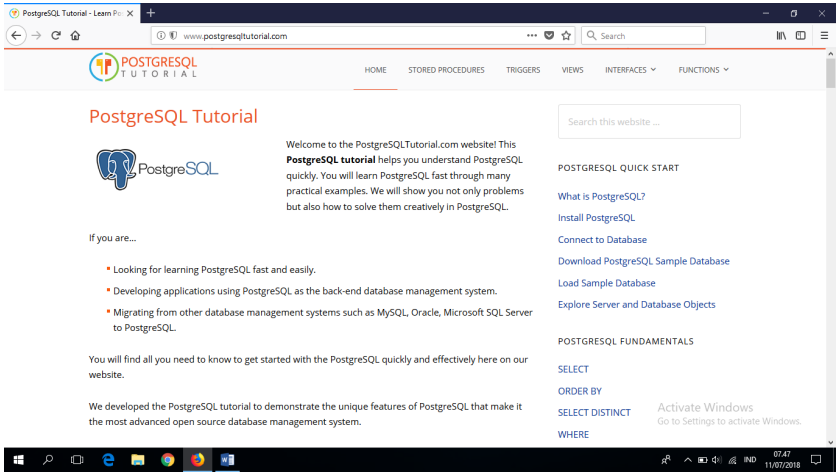
http://workshops.boundlessgeo.com/postgis-intro/

Selain itu, bahan dapat juga diambil dari :

https://www.tutorialspoint.com/postgresql/index.htm

http://www.bostongis.com/?content_name=postgisonline

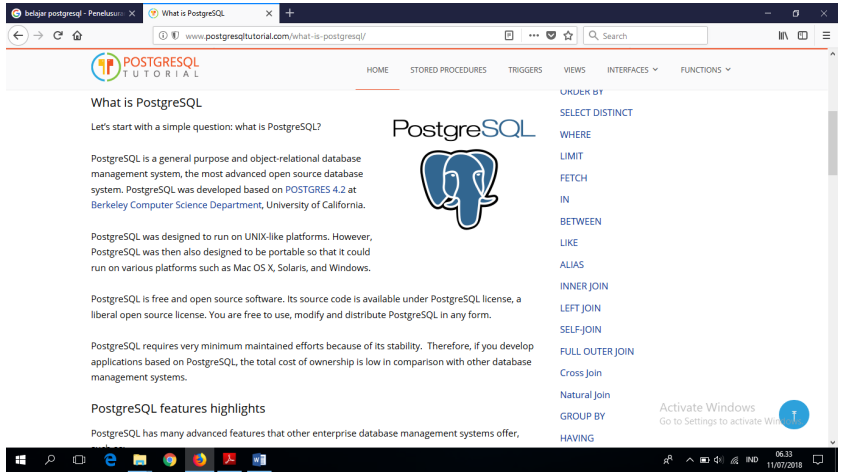
https://postgis.net/documentation/

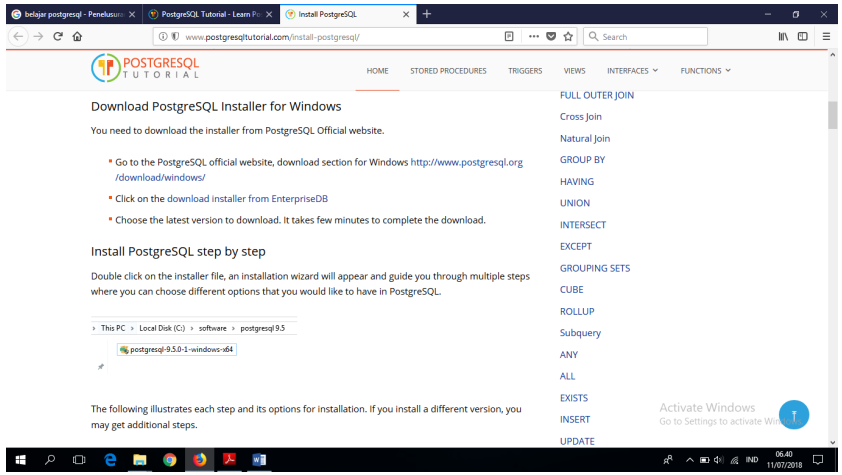


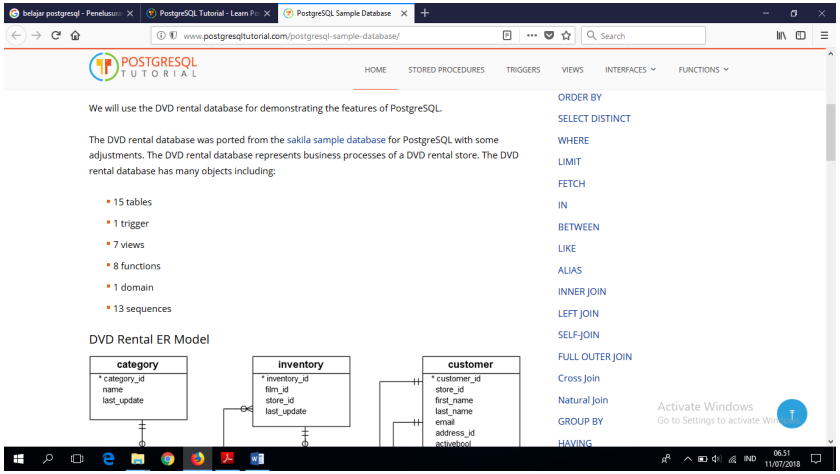
Materi minggu 1:

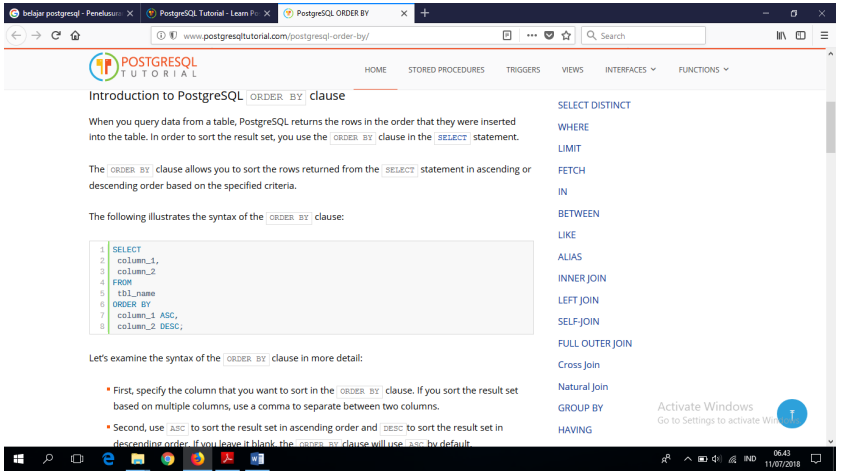
1. Getting started with PostgreSQL

- overview singkat tentang of PostgreSQL
- install PostgreSQL dan konek PostgreSQL database server dari psql atau pgAdmin.
- Download contoh database PostgreSQL dan masukkan ke dalam PostgreSQL database server







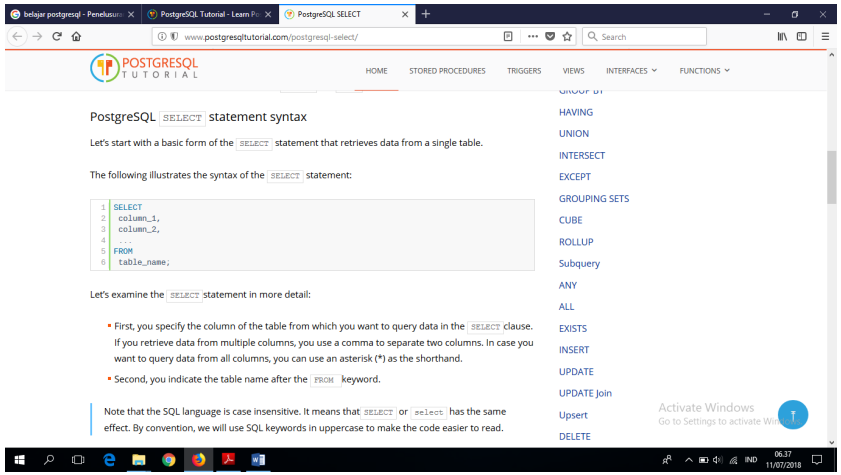


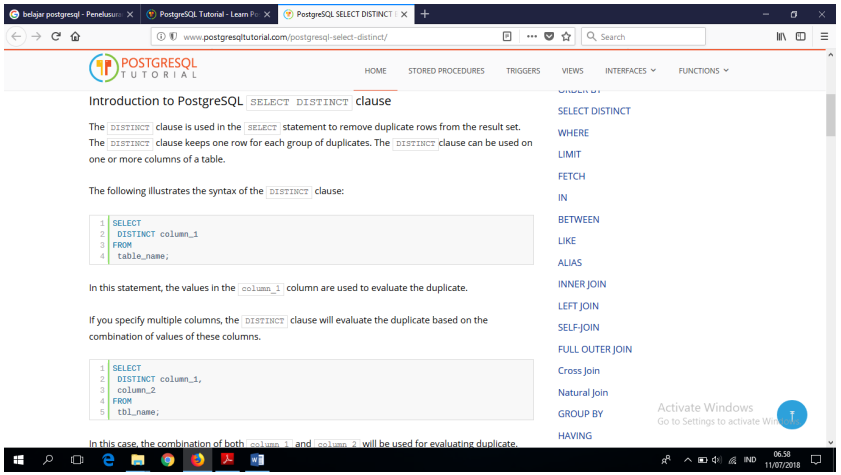
2. Querying Data

Select

Order By

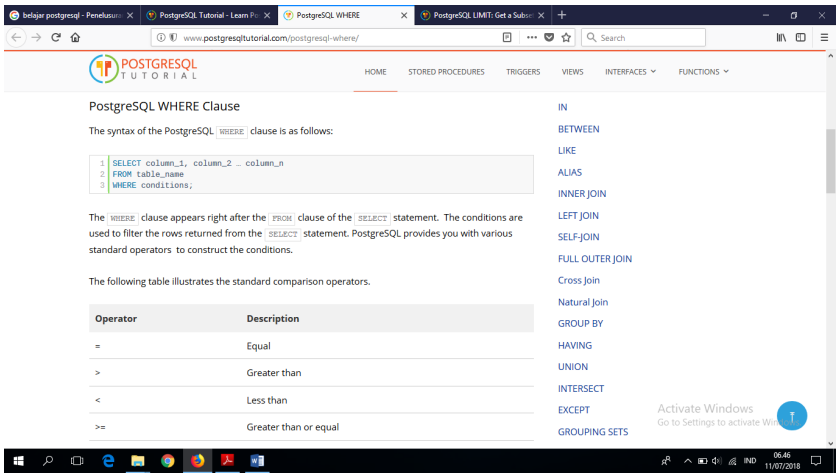
Select Distinct





3. Filtering data

Where
Limit
Fetch
In
Between
Like



PostgreSQL WHERE Clause

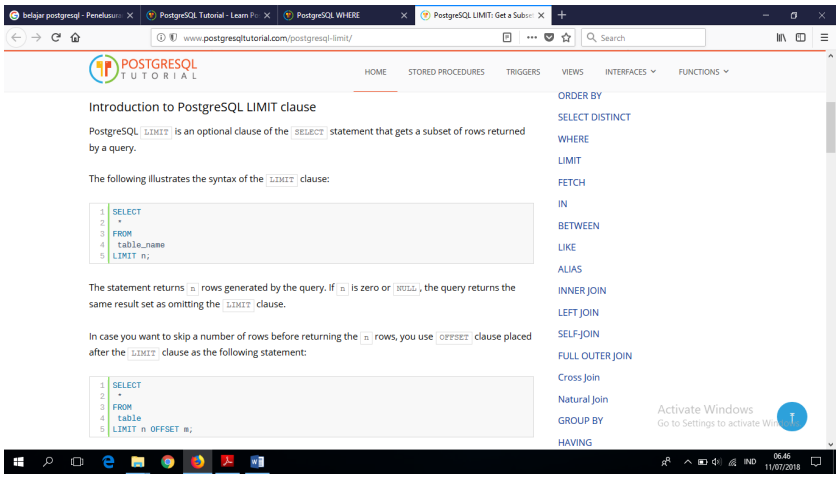
The syntax of the PostgreSQL WHERE clause is as follows:

```
SELECT column_1, column_2 … column_n
FROM table_name
WHERE conditions;
```

The WHERE clause appears right after the FROM clause of the SELECT statement. The conditions are used to filter the rows returned from the SELECT statement. PostgreSQL provides you with various standard operators to construct the conditions.

The following table illustrates the standard comparison operators.

Operator	Description
=	Equal
>	Greater than
<	Less than
>=	Greater than or equal



Introduction to PostgreSQL LIMIT clause

PostgreSQL LIMIT is an optional clause of the SELECT statement that gets a subset of rows returned by a query.

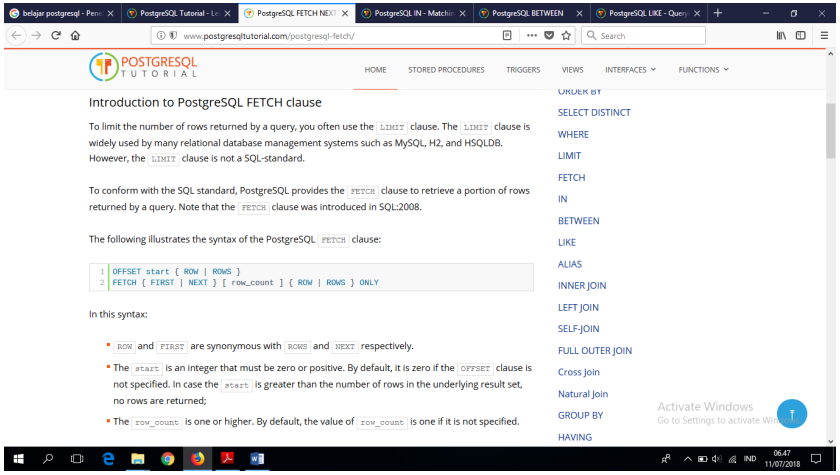
The following illustrates the syntax of the LIMIT clause:

```
SELECT
 *
FROM
 table_name
LIMIT n;
```

The statement returns n rows generated by the query. If n is zero or NULL, the query returns the same result set as omitting the LIMIT clause.

In case you want to skip a number of rows before returning the n rows, you use OFFSET clause placed after the LIMIT clause as the following statement:

```
SELECT
 *
FROM
 table
LIMIT n OFFSET m;
```



Introduction to PostgreSQL FETCH clause

To limit the number of rows returned by a query, you often use the LIMIT clause. The LIMIT clause is widely used by many relational database management systems such as MySQL, H2, and HSQLDB. However, the LIMIT clause is not a SQL-standard.

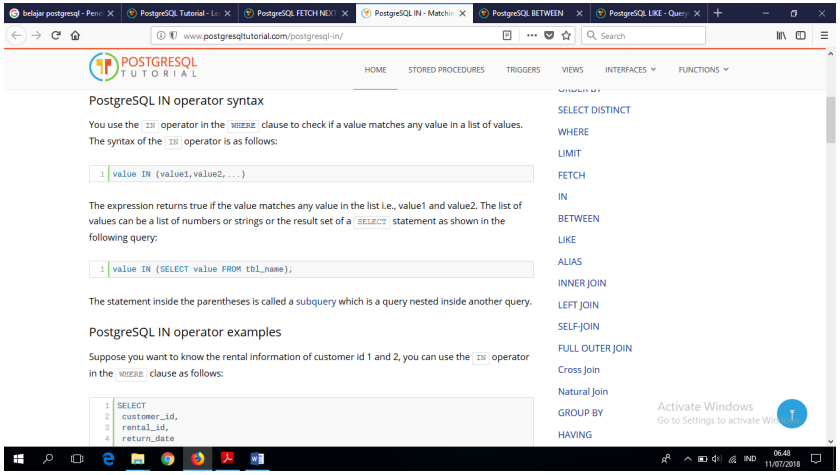
To conform with the SQL standard, PostgreSQL provides the FETCH clause to retrieve a portion of rows returned by a query. Note that the FETCH clause was introduced in SQL:2008.

The following illustrates the syntax of the PostgreSQL FETCH clause:

```
OFFSET start { ROW | ROWS }
FETCH { FIRST | NEXT } [ row_count ] { ROW | ROWS } ONLY
```

In this syntax:

- ROW and FIRST are synonymous with ROWS and NEXT respectively.
- The start is an integer that must be zero or positive. By default, it is zero if the OFFSET clause is not specified. In case the start is greater than the number of rows in the underlying result set, no rows are returned;
- The row_count is one or higher. By default, the value of row_count is one if it is not specified.



PostgreSQL IN operator syntax

You use the IN operator in the WHERE clause to check if a value matches any value in a list of values. The syntax of the IN operator is as follows:

```
value IN (value1,value2,...)
```

The expression returns true if the value matches any value in the list i.e., value1 and value2. The list of values can be a list of numbers or strings or the result set of a SELECT statement as shown in the following query:

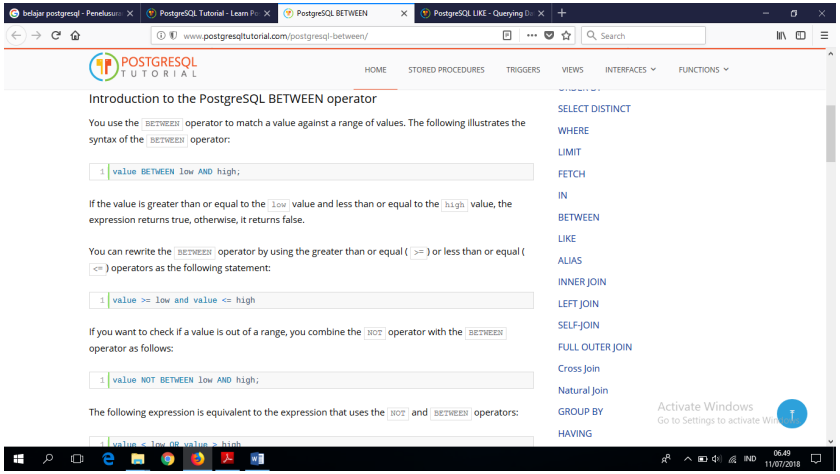
```
value IN (SELECT value FROM tbl_name);
```

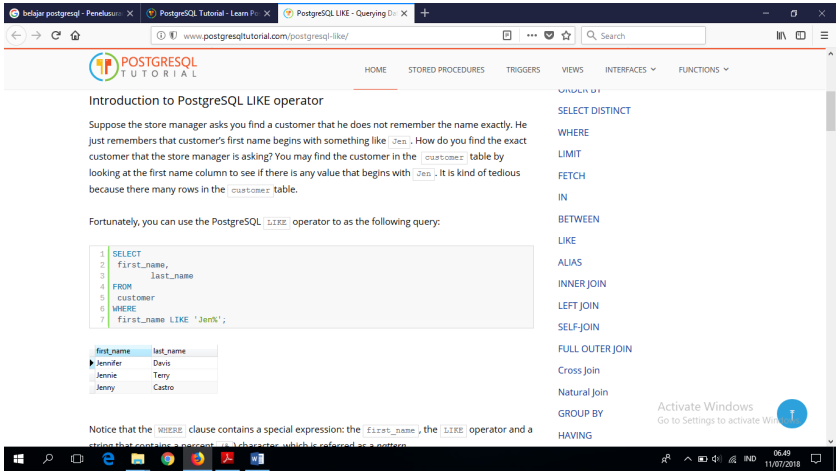
The statement inside the parentheses is called a subquery which is a query nested inside another query.

PostgreSQL IN operator examples

Suppose you want to know the rental information of customer id 1 and 2, you can use the IN operator in the WHERE clause as follows:

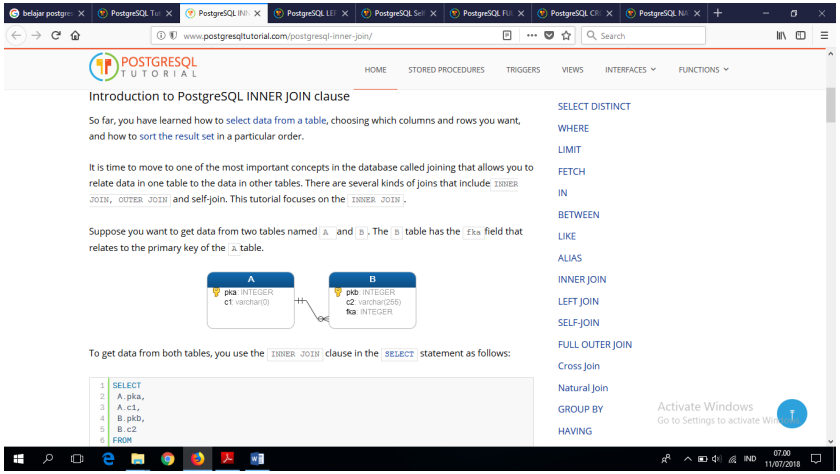
```
SELECT
 customer_id,
 rental_id,
 return_date
```

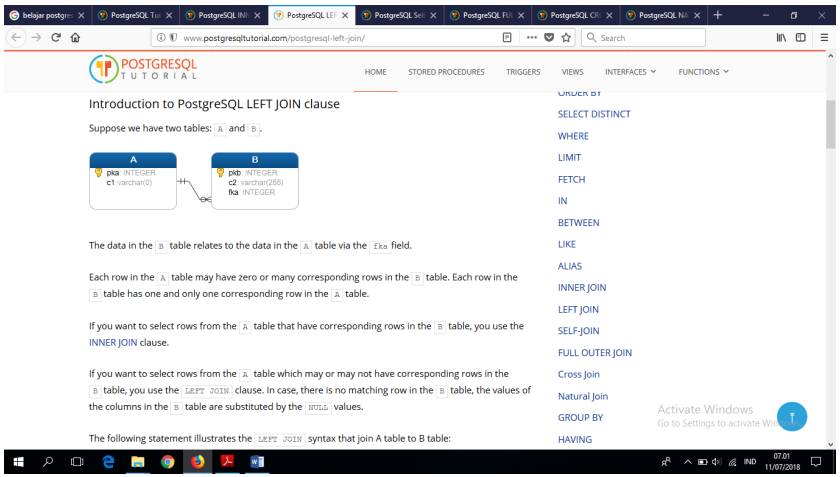




4. Joining multiple tables

Inner Join.
Left Join.
Self-join
Full Outer Join
Cross Join
Natural Join





Introduction to PostgreSQL self-join

A self-join is a query in which a table is joined to itself. Self-joins are useful for comparing values in a column of rows within the same table.

To form a self-join, you specify the same table twice with different aliases, set up the comparison, and eliminate cases where a value would be equal to itself.

The following query shows how to join the table A to itself:

```
SELECT column_list
FROM A a1
INNER JOIN B b1 ON join_predicate;
```

In this syntax, table `A` is joined to itself using the `INNER JOIN` clause. Note that you can also use the `LEFT JOIN` or `RIGHT JOIN` clause.

PostgreSQL self-join examples

1) Querying hierarchy data example

Let's set up a sample table for the demonstration.

Suppose, we have the following organization structure:

Introduction to the PostgreSQL CROSS JOIN clause

A `CROSS JOIN` clause allows you to produce the Cartesian Product of rows in two or more tables. Different from the other JOIN operators such as LEFT JOIN or INNER JOIN, the `CROSS JOIN` does not have any matching condition in the join clause.

Suppose we have to perform the `CROSS JOIN` of two tables T1 and T2. For every row from T1 and T2 i.e., a cartesian product, the result set will contain a row that consists of all columns in the T1 table followed by all columns in the T2 table. If T1 has N rows, T2 has M rows, the result set will have N x M rows.

The following illustrates the syntax of the PostgreSQL CROSS JOIN clause:

```
SELECT *
FROM T1
CROSS JOIN T2;
```

The following statement is also equivalent to the CROSS JOIN above:

```
SELECT *
FROM T1, T2;
```

You can use the INNER JOIN clause with the condition evaluates to true to perform the cross join as follows:

Introduction to the PostgreSQL FULL OUTER JOIN

Suppose, you want to perform a full outer join of two tables: A and B. The following illustrates the syntax of the FULL OUTER JOIN:

```
SELECT * FROM A
FULL [OUTER] JOIN B on A.id = B.id;
```

The `OUTER` keyword is optional.

The full outer join combines the results of both left join and right join. If the rows in the joined table do not match, the full outer join sets NULL values for every column of the table that lacks a matching row. For the matching rows , a single row is included in the result set that contains columns populated from both joined tables.

The following Venn diagram illustrates the FULL OUTER JOIN operation:

A natural join is a join that creates an implicit join based on the same column names in the joined tables. See the following syntax of PostgreSQL natural join:

```
SELECT *
FROM T1
NATURAL [INNER, LEFT, RIGHT] JOIN T2;
```

A natural join can be an inner join, left join, or right join. If you do not specify a join explicitly e.g., `INNER JOIN` , `LEFT JOIN` , `RIGHT JOIN` , PostgreSQL will use the `INNER JOIN` by default.

If you use the asterisk (*) in the select list, the result will contain the following columns:

- All the common columns, which are the columns in the both tables that have the same name
- Every column in the first and second tables that is not a common column

PostgreSQL NATURAL JOIN examples

To demonstrate the PostgreSQL natural join, we will create two tables: `categories` and `products` . The following `CREATE TABLE` statements create the `categories` and `products` table.

```
CREATE TABLE categories (
 category_id serial PRIMARY KEY,
 category_name VARCHAR (255) NOT NULL
);
```

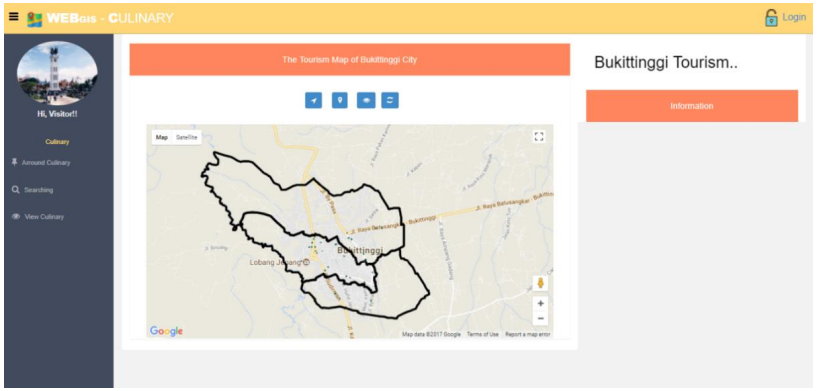
PROGRAM APLIKASI GIS KULINER KHAS BUKITTINGGI

SURYA AFNARIUS DAN FITRIANY CHAIRUNNISA
SISTEM INFORMASI
UNIVERSITAS ANDALAS

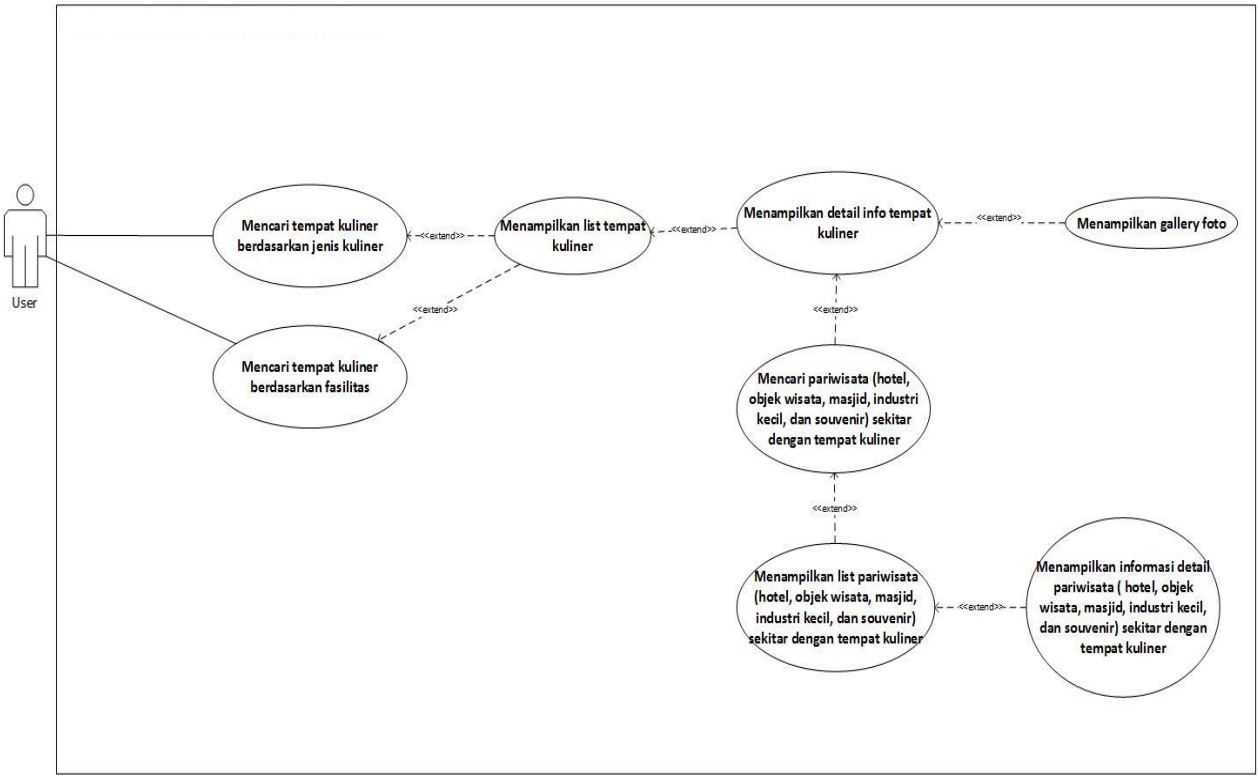
PROGRAM APLIKASI GIS KULINER KHAS BUKITTINGGI

Ada 6 (enam) modul program/ fitur/ fungsional, yaitu modul program untuk:

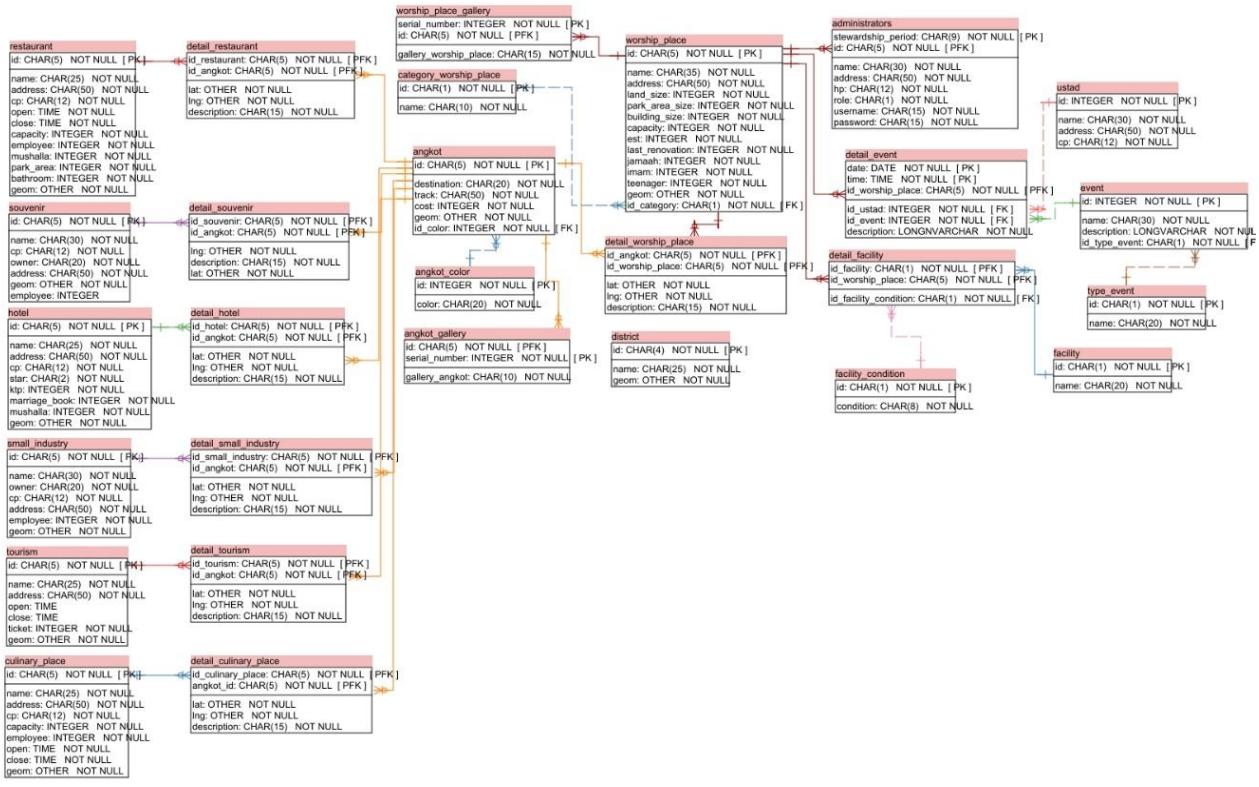
1. Mencari tempat kuliner berdasarkan jenis kuliner
2. Mencari tempat kuliner berdasarkan fasilitas
3. Menampilkan detail informasi tempat kuliner
4. Menampilkan galeri tempat kuliner
5. Menampilkan pariwisata (objek wisata, hotel, restoran, kuliner, industri kecil, dan souvenir) yang berada disekitaran tempat kuliner
6. Menampilkan informasi detail pariwisata



SURYA AFNARIUS DAN FITRIANY CHAIRUNNISA
SISTEM INFORMASI
UNIVERSITAS ANDALAS



Usecase diagram aplikasi GIS kuliner khas Bukittinggi



Ada enam modul program, yaitu:

1. Mencari tempat kuliner berdasarkan jenis kuliner

a. Use case scenario

Table 1. Skenario Mencari Tempat Kuliner Berdasarkan Jenis Kuliner

Use case name	Mencari Tempat Kuliner Berdasarkan Jenis Kuliner
Participating Actor	*User*
Flow Of Events	1. *User* memilih pada menu *sidebar button view culinary*
	2. *User* memilih *select culinary* pada sub menu *button view culinary*
	3. *User* memilih jenis kuliner yang tersedia dalam bentuk *checkbox*
	4. *User* menekan *button search*
	5. Sistem menampilkan posisi dan daftar tempat kuliner yang dicari
Exit Condition	*User* dapat melihat tempat kuliner berdasarkan jenis kuliner
Alternative Course	1. *User* memilih pada menu *sidebar button view culinary*
	2. *User* memilih *select culinary* pada sub menu *button view culinary*
	3. *User* memilih jenis kuliner yang tersedia dalam bentuk *checkbox*
	4. *User* menekan *button search*
	5. Sistem akan menampilkan "Data Didn't Exist!"

b. Code

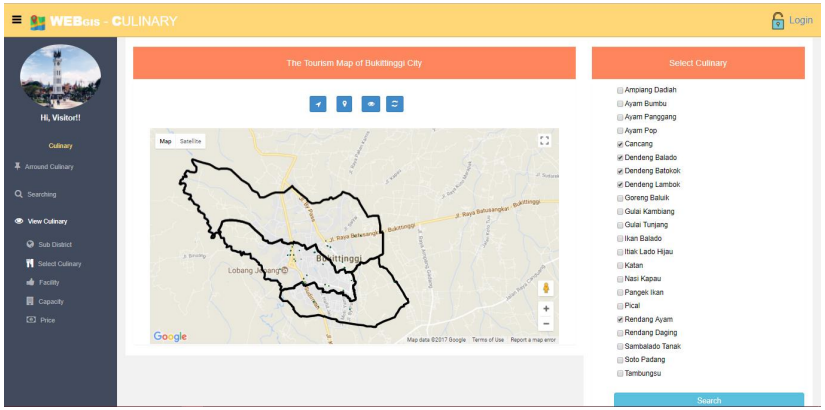
- Javascript

```
function viewkull()
{
  clearroute2();
  $("#hasilik").show();
  $('#hasilcari1').show();
  $('#hasilcari').empty();
  $('#hasilcari').empty();
  $('#hasilpencarian').empty();

  hapusInfo();
  hapusMarkerTerdekat();
  var fas=selectkul.value;
  var arrayLay=[];
  for(i=0;i<$("input[name=culinary]:checked").length;i++){
    arrayLay.push($("input[name=culinary]:checked")[i].value);
  } //Perulangan saat memilih checkbox
  console.log('zz');
  if (arrayLay==''){
    alert('Pilih Kuliner');
  }else{
    $('#hasilcari').append("<thead><th>Name</th><th
        colspan='3'>Action</th></thead>");
    $.ajax({ url: server+'selectkul.php?lay='+arrayLay, data: "",
        dataType: 'json', success: function(rows){
      console.log("hai");
      if(rows==null)//Jika tidak ada data
          {
            alert('Data not found');
          }
        for (var i in rows)
          {
            var row     = rows[i];
            var id    = row.id;
            var nama_kuliner    = row.name;
            var nama_tempat_kuliner    = row.name;
            var latitude  = row.latitude ;
            var longitude = row.longitude ;
            centerBaru    =    new    google.maps.LatLng(latitude,
        longitude);
            marker = new google.maps.Marker
          ({
            position: centerBaru,
            icon:'assets/img/cul.png',
            map: map,
            animation: google.maps.Animation.DROP,
          });
            console.log(name);
            console.log(latitude);
            console.log(longitude);
            markersDua.push(marker);
            map.setCenter(centerBaru);
            map.setZoom(16);
            clickMarker(centerBaru, id);

        $('#hasilcari').append("<tr><td>"+nama_tempat_kuliner+"</t
        d><td><a     role='button'     class='btn     btn-success'
        onclick='detculi(\""+id+"\");detailinfokul(\""+id+"\");'>S
        how</a></td><td><a role='button' class='btn btn-danger fa
        fa-taxi' onclick='kulAngkot(\""+id+"\")'></a></td></tr>");
          }
          $('#hasilpencarian').append("<h5      class='box-title'
        id='hasilpencarian'>Result :</h5>"+rows.length);
    }});
  }
}
```

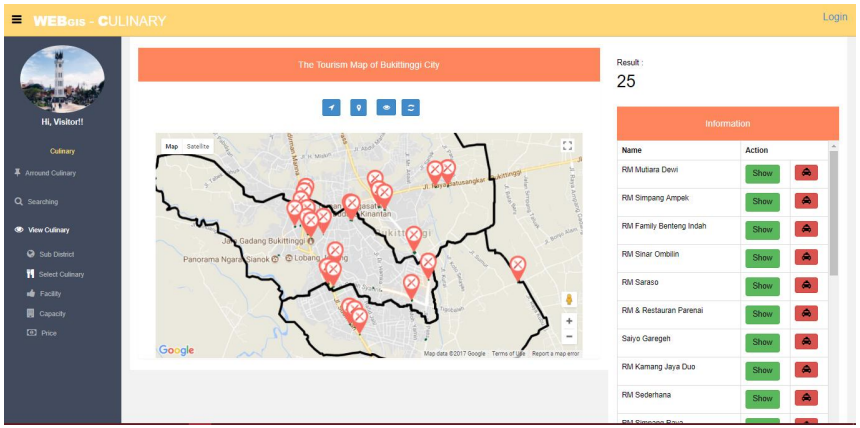
- selectcul.php

```
<?php
require 'connect.php';

$lay=$_GET['lay'];
$lay =explode(",", $lay);
$c = "";
for($i=0;$i<count($lay);$i++){ //Perulangan saat memilih checkbox
     if($i == count($lay)-1){
          $c .= "'".$lay[$i]."'";
     }else{
          $c .= "'".$lay[$i]."',";
     }
}
$querysearch="select
culinary_place.id,culinary_place.name,ST_X(ST_Centroid(culinary_pl
ace.geom)) AS lng, ST_Y(ST_CENTROID(culinary_place.geom)) As lat
from culinary_place join detail_culinary on
culinary_place.id=detail_culinary.id_culinary_place where
detail_culinary.id_culinary in ($c) group by id";
$hasil=pg_query($querysearch);
while($row = pg_fetch_array($hasil))
     {
          $id=$row['id'];
          $name=$row['name'];
          $longitude=$row['lng'];
          $latitude=$row['lat'];
     $dataarray[]=array('id'=>$id,'name'=>$name,'longitude'=>$long
itude,'latitude'=>$latitude);
     }
echo json_encode ($dataarray);
?>
```

c. Contoh luaran program



Memilih Jenis Kuliner dalam Checkbox



Tempat Kuliner Berdasarkan Jenis Kuliner

2. Mencari tempat kuliner berdasarkan fasilitas

a. Use case scenario

Tabel 2. Skenario Mencari Tempat Kuliner Berdasarkan Fasilitas

Use case name	Mencari Tempat Kuliner Berdasarkan Fasilitas
Participating Actor	*User*
Flow Of Events	1. *User* memilih pada menu *sidebar button view culinary*
	2. *User* memilih *facility* pada sub menu *button view culinary*
	3. *User* memilih jenis fasilitas yang tersedia dalam bentuk *checkbox*
	4. *User* menekan *button search*
	5. Sistem menampilkan posisi dan daftar tempat kuliner yang dicari
Exit Condition	*User* dapat melihat tempat kuliner berdasarkan jenis kuliner
Alternative Course	6. *User* memilih pada menu *sidebar button view culinary*

	7. *User* memilih *facility* pada sub menu *button view culinary*
	8. *User* memilih jenis fasilitas yang tersedia dalam bentuk *checkbox*
	9. *User* menekan *button search*
	10. Sistem akan menampilkan "Data Didn't Exist!"

b. Code

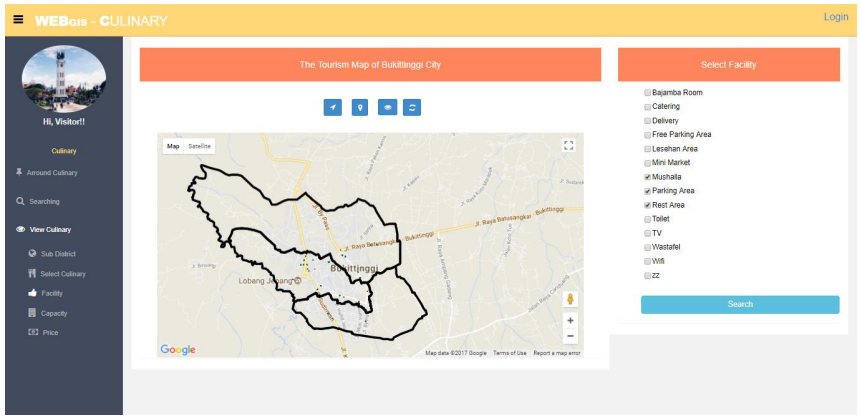
- Javascript

```
function viewfas()
{
  clearroute2();
  $("#hasilik").show();
  $('#hasilcari1').show();
  $('#hasilcari').empty();
  $('#hasilcari').empty();
  $('#hasilpencarian').empty();
      hapusInfo();
      hapusMarkerTerdekat();
  var fas=selectfacility.value;
  var arrayLay=[];
  for(i=0;i<$("input[name=facility]:checked").length;i++){
    arrayLay.push($("input[name=facility]:checked")[i].value);
  } //Perulangan saat memilih checkbox
  console.log(arrayLay);
  if (arrayLay==''){ //Jika Fasilitas Kosong
    alert('Pilih Fasilitas');
  }else{
    $('#hasilcari').append("<thead><th>Name</th><th
colspan='3'>Action</th></thead>");
    $.ajax({  url:  server+'selectfas.php?lay='+arrayLay,  data:  "",
dataType: 'json', success: function(rows){
      console.log("hai");
      if(rows==null) //Jika Data Kosong
            {
              alert('Data not found');
            }
        for (var i in rows)
            {
              var row     = rows[i];
              var id   = row.id;
              var id_facility   = row.id;
              var name_facility   = row.name;
              var nama_tempat_kuliner   = row.name;
              var latitude  = row.latitude ;
              var longitude = row.longitude ;
              centerBaru = new google.maps.LatLng(latitude, longitude);
              marker = new google.maps.Marker
            ({
              position: centerBaru,
              icon:'assets/img/cul.png',
              map: map,
              animation: google.maps.Animation.DROP,
            });
              console.log(name);
              console.log(latitude);
              console.log(longitude);
              markersDua.push(marker);
              map.setCenter(centerBaru);
              map.setZoom(12);
              clickMarker(centerBaru, id);

  $('#hasilcari').append("<tr><td>"+nama_tempat_kuliner+"</td><td><a
  role='button'                class='btn                btn-success'
  onclick='detculi(\""+id+"\");detailinfokul(\""+id+"\");'>Show</a></td>
  <td><a    role='button'    class='btn    btn-danger    fa    fa-taxi'
  onclick='kulAngkot(\""+id_facility+"\")'></a></td></tr>");
            }
             $('#hasilpencarian').append("<h5         class='box-title'
  id='hasilpencarian'>Result :</h5>"+rows.length);
    }});
  }
}
```

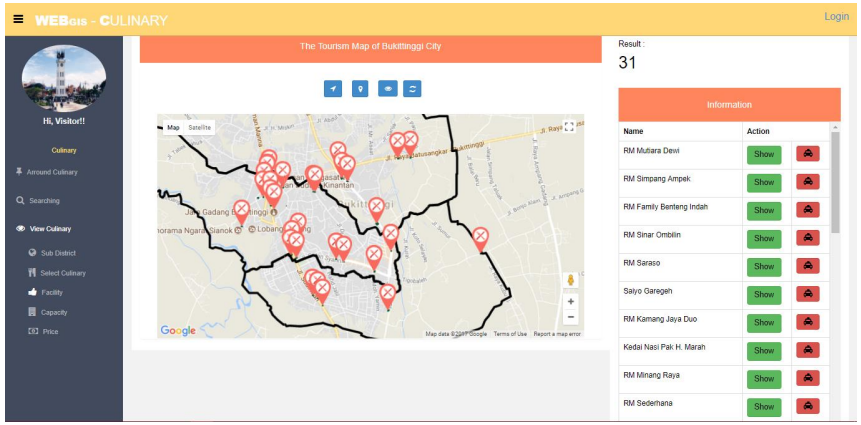
- selectfas.php

```
<?php
require 'connect.php';
$lay=$_GET['lay'];
$lay = explode(",", $lay);
$c = "";
for($i=0;$i<count($lay);$i++){
    if($i == count($lay)-1){
          $c .= "'".$lay[$i]."'";
    }else{
          $c .= "'".$lay[$i]."',";
    }
}
$querysearch="select
culinary_place.id,culinary_place.name,ST_X(ST_Centroid(culinary_place.g
eom))  AS  lng,  ST_Y(ST_CENTROID(culinary_place.geom))  As  lat  from
culinary_place         join         detail_facility_culinary         on
culinary_place.id=detail_facility_culinary.id_culinary_place      where
detail_facility_culinary.id_facility in ($c) group by id";
$hasil=pg_query($querysearch);
while($row = pg_fetch_array($hasil))
    {
          $id=$row['id'];
          $id_facility=$row['id'];
          $name=$row['name'];
          //$name=$row['name'];
          $longitude=$row['lng'];
          $latitude=$row['lat'];

    $dataarray[]=array('id'=>$id,'id_facility'=>$id,'name'=>$name,'long
itude'=>$longitude,'latitude'=>$latitude);
    }
echo json_encode ($dataarray);
?>
```

c. Contoh luaran program



Memilih Fasilitas dalam Checkbox



Tempat Kuliner Berdasarkan Fasilitas

3. Menampilkan Detail Informasi Tempat Kuliner

a. Use case scenario

Tabel 3. Skenario Menampilkan Detail Informasi Tempat Kuliner

Use case name	Menampilkan Detail Informasi Tempat Kuliner
Participating Actor	*User*
Flow Of Events	1. *User* menekan *button show* pada list tempat kuliner yang diinginkan
	2. Sistem menampilkan detail tempat kuliner yang dipilih
Exit Condition	*User* dapat melihat detail informasi tempat kuliner yang dipilih

b. Code

- Javascript

```
function detailinfokul(id144)
{
$('#info').empty();
$('#hasilcaridetculi').empty();
$('#hasilcaridetculi1').show();
$('#hasildet').show();
$('#hasilcaridet').empty();
$('#hasilcaridet1').show();
   hapusInfo();
       $.ajax({
      url:  server+'detailinfokul.php?info='+id144,  data:  "",  dataType:
   'json', success: function(rows)
        {
         for (var i in rows) //Perulangan menampilkan data
          {
            console.log('ddd');
            var row = rows[i];
            var id = row.id;
            var namaa = row.name;
            var capacity = row.capacity;
            var address=row.address;
            var cp=row.cp;
            var open = row.open;
            var close = row.close;
            var fasilitas = row.fasilitas;
            var latitude  = row.latitude; ;
            var longitude = row.longitude ;
            centerBaru     =     new     google.maps.LatLng(row.latitude,
   row.longitude);
           marker = new google.maps.Marker
           ({
             position: centerBaru,
             icon:'assets/img/cul.png',
             map: map,
             animation: google.maps.Animation.DROP,
```

```
            });
              console.log(latitude);
              console.log(longitude);
              markersDua.push(marker);
            map.setCenter(centerBaru);
            map.setZoom(18);
    <br> Kapasitas : "+kapasitas+"");
          }
        }
      });
}

function detculi(id14433){

  $('#info').empty();
  $('#hasilcaridetculi').empty();
  $('#hasilcaridetculi1').show();
  $('#hasildet').show();
  $('#hasilcaridet').empty();
  $('#hasilcaridet1').show();
   hapusInfo();
       $.ajax({
      url: server+'detculi.php?info='+id14433, data: "", dataType:
  'json', success: function(rows)
        {
          $('#hasilcaridet').append("<tr><td
  colspan='2'><strong>Culinary</strong></td><td><strong>Price</st
  rong></td></tr>");
        for (var i in rows) //Perulangan menampilkan data
          {
            var row = rows[i];
            var id = row.id;
            var namaa = row.name;
            var capacity = row.capacity;
            var address=row.address;
            var cp=row.cp;
            var open=row.open;
            var close=row.close;
            var price = row.price;
            var culinary = row.culinary;
            var latitude  = row.latitude; ;
            var longitude = row.longitude ;
            centerBaru   =   new   google.maps.LatLng(row.latitude,
  row.longitude);
            marker = new google.maps.Marker
            ({
              position: centerBaru,
              icon:'assets/img/cul.png',
              map: map,
              animation: google.maps.Animation.DROP,
            });
              console.log(latitude);
              console.log(longitude);
              console.log(id);
              markersDua.push(marker);
            map.setCenter(centerBaru);
            map.setZoom(18);
```

```
            infowindow = new google.maps.InfoWindow({
            position: centerBaru,
            content:                                    "<center><span
  style=color:black><b>Information</b><table><tr><td><i  class='fa
  fa-home'></i>Nama</td><td>:</td><td>
  "+namaa+"</td></tr><br><tr><td><i        class='fa        fa-map-
  marker'></i>Alamat</td><td>:</td><td>
  "+address+"</td></tr><br><tr><td><i         class='fa         fa-
  phone'></i>Telepon</td><td>:</td><td>
  "+cp+"</td></tr><br><tr><td><i        class='fa        fa-clock-
  o'></i>Open</td><td>:</td><td>  "+open+"</td></tr><br><tr><td><i
  class='fa                fa-clock-o'></i>Close</td><td>:</td><td>
  "+close+"</td></tr><br><tr><td><i          class='fa          fa-
  building'></i>Capacity</td><td>:</td><td>
  "+capacity+"</td></tr></table></span><br><input    type='button'
  class='btn       btn-success'       value='Object       Arround'
  onclick='tampil_sekitar(\""+latitude+"\",\""+longitude+"\",\""+
  namaa+"\")'<br>  <input  type='button'  class='btn  btn-
  success'                                              value='Gallery'
  onclick='gallery(\""+id+"\")'<br>   <input type='button'
  class='btn              btn-success'              value='Route'
  onclick='callRoute(centerLokasi,centerBaru);rutetampil()' />",
            pixelOffset: new google.maps.Size(0, -33)
            });
          infoposisi.push(infowindow);
          hapusInfo();
          infowindow.open(map);

          }
        }
      });
}
```

- detailinfokul.php

```
<?php
require 'connect.php';
$info = $_GET["info"];
$querysearch                              =                              "select
culinary_place.id,culinary_place.name,culinary_place.capacity,culinary_pl
ace.address,culinary_place.cp,culinary_place.open,culinary_place.close,
newtable.id_culinary_place,ST_X(ST_Centroid(culinary_place.geom)) AS lng,
ST_Y(ST_CENTROID(culinary_place.geom))  As  lat,  newtable.fasilitas  from
(select                          detail_facility_culinary.id_culinary_place,
string_agg(facility_culinary.facility,    ',    ')    as    fasilitas    from
detail_facility_culinary join facility_culinary on facility_culinary.id =
detail_facility_culinary.id_facility                                    where
detail_facility_culinary.id_culinary_place='$info'        group        by
detail_facility_culinary.id_culinary_place)     as     newtable     join
culinary_place   on   culinary_place.id   =   newtable.id_culinary_place";
//Menampilkan data untuk detail informasi


$hasil=pg_query($querysearch);
while($row = pg_fetch_array($hasil)) //Perulangan menampilkan data
      {
              $id=$row['id'];
```

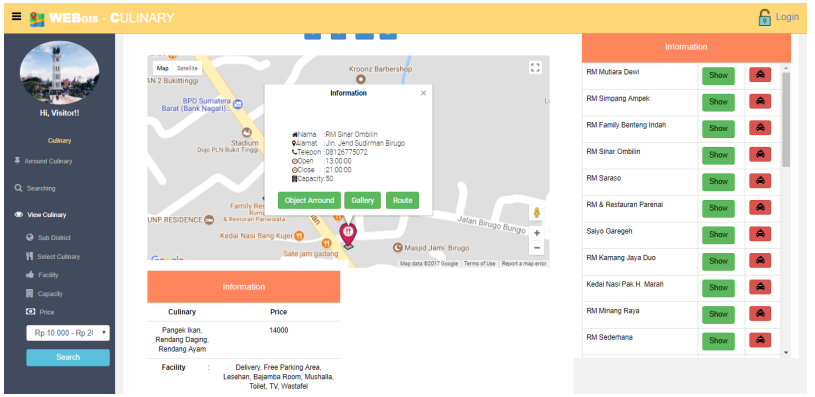
```
                $name=$row['name'];
                $capacity=$row['capacity'];
                $address=$row['address'];
                $cp=$row['cp'];
                $open=$row['open'];
                $close=$row['close'];
                $fasilitas=$row['fasilitas'];
                $longitude=$row['lng'];
                $latitude=$row['lat'];


$dataarray[]=array('id'=>$id,'name'=>$name,'capacity'=>$capacity,'address
'=>$address,'cp'=>$cp,'close'=>$close,'fasilitas'=>$fasilitas,'open'=>$op
en,'longitude'=>$longitude,'latitude'=>$latitude);
        }
echo json_encode ($dataarray);
?>
```

- detculi.php

```
<?php
require 'connect.php';
$info = $_GET["info"];


$querysearch                                                    ="select
culinary_place.id,culinary_place.name,culinary_place.capacity,culinary_pl
ace.address,culinary_place.cp,culinary_place.open,culinary_place.close,
newtable.id_culinary_place,newtable.price,ST_X(ST_Centroid(culinary_place
.geom))    AS    lng,    ST_Y(ST_CENTROID(culinary_place.geom))    As    lat,
newtable.culinary                        from                        (select
detail_culinary.id_culinary_place,detail_culinary.price,
string_agg(culinary.name,  ',  ')  as  culinary  from  detail_culinary  join
culinary    on    culinary.id    =    detail_culinary.id_culinary    where
detail_culinary.id_culinary_place='$info'              group              by
detail_culinary.id_culinary_place,detail_culinary.price) as newtable join
culinary_place on culinary_place.id = newtable.id_culinary_place";
//Menampilkan data untuk detail informasi



$hasil=pg_query($querysearch);
while($row = pg_fetch_array($hasil)) //Perulangan menampilkan data
      {
                $id=$row['id'];
                $name=$row['name'];
                $capacity=$row['capacity'];
                $address=$row['address'];
                $cp=$row['cp'];
                 $open=$row['open'];
                  $close=$row['close'];
                $price=$row['price'];
                $culinary=$row['culinary'];
                $longitude=$row['lng'];
                $latitude=$row['lat'];

$dataarray[]=array('id'=>$id,'name'=>$name,'capacity'=>$capacity,'address
'=>$address,'cp'=>$cp,'open'=>$open,'close'=>$close,'culinary'=>$culinary
,'price'=>$price,'longitude'=>$longitude,'latitude'=>$latitude);
      }
```

```
echo json_encode ($dataarray);
?>
```

c. Contoh luaran program



Detail informasi tempat kuliner yang dipilih

4. Menampilkan Gallery Foto

a. Use case scenario

Tabel 4. Skenario Menampilkan Gallery Foto

Use case name	Menampilkan Gallery Foto
Participating Actor	*User*
Flow Of Events	1. *User* menekan *button show* pada objek yang dipilih
	2. Sistem menampilkan detail informasi objek yang dipilih
	3. *User* menekan *button gallery* pada objek yang dipilih
	4. Sistem menampilkan gallery foto objek yang dipilih
Exit Condition	*User* dapat melihat gallery foto

b. Code

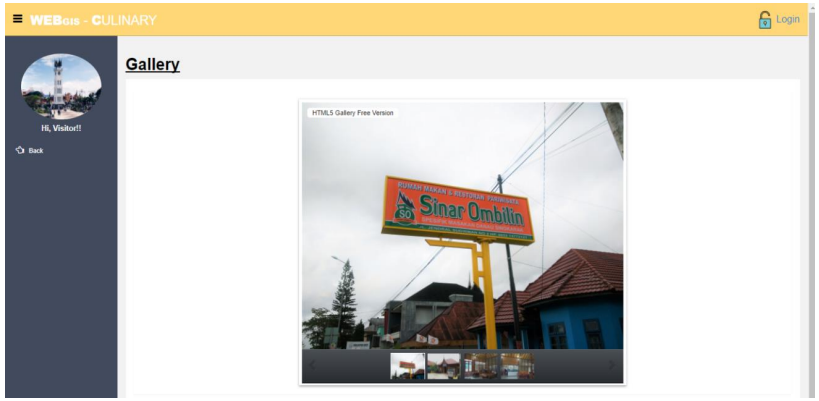
- Javascript

```
function gallery(azz){
      console.log(azz);
    window.open(server+'gallery.php?idgallery='+azz);
   }
```

- gallery.php

```
<?php
            require 'connect.php';

            $id = $_GET["idgallery"];

              $querysearch  ="   SELECT a.id, b.gallery_culinary as foto
                                 FROM  culinary_place  as  a  left  join
culinary_gallery as b on a.id=b.id where a.id='$id' "; //Menampilkan foto
      echo "zzz";
               $hasil=pg_query($querysearch);
               while($baris   =   pg_fetch_assoc($hasil))   //Perulangan
menampilkan foto
                                  {
                                    if(($baris['foto']=='-
')||($baris['foto']=='')) //Jika foto tidak ada dalam database
                                    {
                                        echo                          "<a
href='img/noimage.jpg'><img src='img/noimage.jpg' ></a>";
                                    }
                                    else//Jika foto ada dalam database
                                    {
                                            echo                          "<a
href='img/".$baris['foto']."'><img src='img/".$baris['foto']."' ></a>";
                                    }

                                  } ?>
```

c. Contoh luaran program



Gallery Tempat Kuliner

5. Melihat pariwisata (objek wisata, hotel, restoran, kuliner, industri kecil, dan souvenir) yang berada disekitaran tempat kuliner yang dipilih)

a. Use case scenario

Tabel 5. Skenario Mencari Souvenir Sekitar dengan Tempat Kuliner

Use case name	Mencari Hotel, Objek Wisata, Masjid, Industri Kecil, dan Souvenir Sekitar dengan Tempat Kuliner
Participating Actor	*User*
Flow Of Events	1. *User* menekan *button show* pada list tempat kuliner yang diinginkan
	2. Sistem menampilkan detail tempat kuliner yang dipilih
	3. *User* menekan *button object arround* pada info window
	4. Sistem menampilkan pilihan objek dalam bentuk *checkbox* dan radius dalam bentuk *seekbar*
	5. *User* memilih objek dan radius yang diinginkan
	6. Sistem menampilkan posisi dan list objek yang dipilih
Exit Condition	*User* dapat melihat hotel, objek wisata, masjid, industri kecil, dan souvenir sekitar dengan tempat kuliner sesuai dengan objek dan radius yang dipilih

b. Code

- Javascript

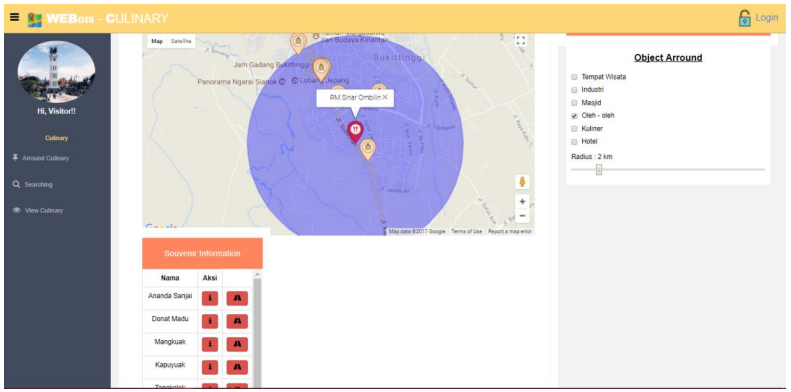
```
function oleholeh_sekitar(latitude,longitude,rad){ // OLEH-OLEH SEKITAR

          $('#hasilcarisouv').empty();
           $('#hasilcarisouv1').show();
          $('#hasilcarisouv').append("<thead><th
     class='centered'>Nama</th><th                                colspan='3'
     class='centered'>Aksi</th></thead>");
          $.ajax({url:
     server+'_sekitar_oleholeh.php?lat='+latitude+'&long='+longitude+'&r
     ad='+rad, data: "", dataType: 'json', success: function(rows){
             for (var i in rows){ //Perulangan menampilkan data
               var row = rows[i];
               var id = row.id;
               var name = row.name;
               var owner = row.owner;
               var cp = row.cp;
```

```
                var address = row.address;

                var lat=row.latitude;
                var lon = row.longitude;

                //POSISI MAP
                centerBaru = new google.maps.LatLng(lat, lon);
                map.setCenter(centerBaru);
                map.setZoom(16);
                var marker = new google.maps.Marker({
                  position: centerBaru,
                  icon:'assets/img/souv.png',
                  animation: google.maps.Animation.DROP,
                  map: map
                  });
                markersDua.push(marker);
                map.setCenter(centerBaru);

                $('#hasilcarisouv').append("<tr><td>"+name+"</td><td><a
      role='button'      class='btn      btn-danger      fa      fa-info'
      onclick='detailinfosou(\""+id+"\")'></a></td><td><a   role='button'
      class='btn            btn-danger            fa            fa-road'
      onclick='route_sekitar(\""+latitude+"\",\""+longitude+"\",\""+lat+"
      \",\""+lon+"\")'></a></td><td><a role='button' class='btn btn-danger
      fa fa-taxi' onclick='souangkot(\""+id+"\")'></a></td></tr>");
             }//end for
           });//end ajax
        }
```

- _sekitar_oleholeh.php

```
<?php

      include('connect.php');
    $latit = $_GET['lat'];
    $longi = $_GET['long'];
      $rad=$_GET['rad'];

      $querysearch="SELECT     id,     name,     owner,     cp,     address,
st_x(st_centroid(geom))    as    lng,    st_y(st_centroid(geom))    as    lat,
st_distance_sphere(ST_GeomFromText('POINT(".$longi."     ".$latit.")',-1),
geom)        as        jarak        FROM              souvenir        where
st_distance_sphere(ST_GeomFromText('POINT(".$longi."     ".$latit.")',-1),
geom) <= ".$rad.""; //Menampilkan data oleh-oleh sesuai radius yang dipilih

      $hasil=pg_query($querysearch);

        while($baris  =  pg_fetch_array($hasil))  //Perulangan  menampilkan
data
            {
                $id=$baris['id'];
                $name=$baris['name'];
                $owner=$baris['owner'];
                $cp=$baris['cp'];
                $address=$baris['address'];

                $latitude=$baris['lat'];
                $longitude=$baris['lng'];

$dataarray[]=array('id'=>$id,'name'=>$name,'cp'=>$cp,'address'=>$address,
"latitude"=>$latitude,"longitude"=>$longitude);
            }
            echo json_encode ($dataarray);

?>
```

c. Contoh luaran program



Pariwisata (objek wisata,hotel, restoran, kuliner, industri kecil, dan souvenir) yang berada disekitaran tempat kuliner yang dipilih

6. Melihat informasi detail pariwisata yang dipilih

a. Use case scenario

Tabel 6. Skenario Menampilkan Detail Informasi Souvenir Sekitar Tempat Kuliner

Use case name	Menampilkan Detail Informasi Hotel, Objek Wisata, Masjid, Industri Kecil, dan Souvenir Sekitar Tempat Kuliner
Participating Actor	*User*
Flow Of Events	1. *User* menekan *button object arround* pada info window
	2. Sistem menampilkan pilihan objek dalam bentuk *checkbox* dan radius dalam bentuk *seekbar*
	3. *User* memilih objek dan radius yang diinginkan
	4. Sistem menampilkan posisi dan list objek yang dipilih
	5. *User* menekan *button show* pada objek yang dipilih

	6. Sistem menampilkan detail informasi objek yang dipilih
Exit Condition	*User* dapat melihat detail informasi hotel, objek wisata, masjid, industri kecil, dan souvenir sekitar dengan tempat kuliner sesuai dengan objek yang dipilih

b. Code

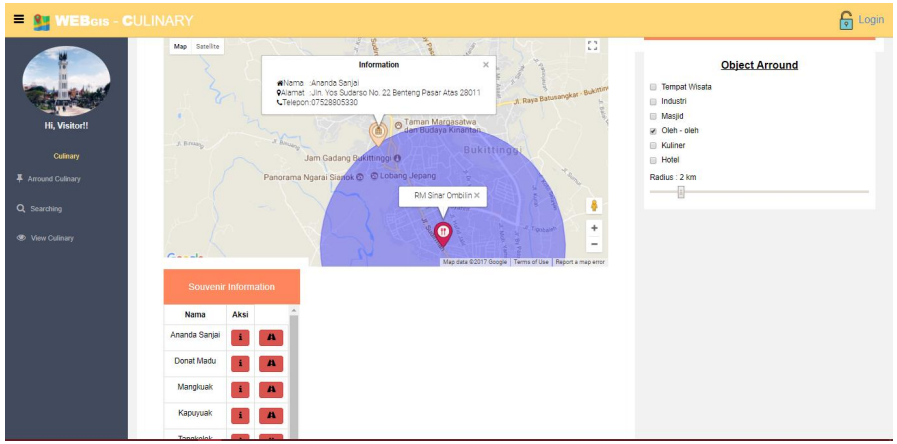
- Javascript

```
function detailinfosou(id14)
{
  $('#info').empty();
   hapusInfo();
      // clearroute2();
      hapusMarkerTerdekat();
       $.ajax({
      url:    server+'detailinfosou.php?info='+id14,    data:    "",
     dataType: 'json', success: function(rows)
        {
         for (var i in rows) //Perulangan menampilkan data
          {
            console.log('ddd');
            var row = rows[i];
            var id = row.id;
            var namaa = row.name;
            var address=row.address;
            var cp = row.cp;
            var owner = row.owner;


            var latitude  = row.latitude; ;
            var longitude = row.longitude ;
            centerBaru   =   new   google.maps.LatLng(row.latitude,
     row.longitude);
           marker = new google.maps.Marker
           ({
             position: centerBaru,
             icon:'assets/img/souv.png',
             map: map,
             animation: google.maps.Animation.DROP,
           });
             console.log(latitude);
             console.log(longitude);
             markersDua.push(marker);
           map.setCenter(centerBaru);
           map.setZoom(18);
           if (address==null)
                  {
                    address="tidak ada";
                  }
              infowindow = new google.maps.InfoWindow({
              position: centerBaru,
              content:                                    "<center><span
      style=color:black><b>Information</b><table><tr><td><i
      class='fa                fa-home'></i>Nama</td><td>:</td><td>
      "+namaa+"</td></tr><br><tr><td><i      class='fa      fa-map-
      marker'></i>Alamat</td><td>:</td><td>
      "+address+"</td></tr><br><tr><td><i       class='fa       fa-
      phone'></i>Telepon</td><td>:</td><td>
      "+cp+"</td></tr></table></span>",
             pixelOffset: new google.maps.Size(0, -33)
              });
           infoposisi.push(infowindow);
           hapusInfo();
           infowindow.open(map);
           }
        }
       });
}
```

- detailinfosou.php

```
<?php
require 'connect.php';
$info = $_GET["info"];
$querysearch ="select id, name, owner, address, cp,ST_X(ST_Centroid(geom))
AS lng, ST_Y(ST_CENTROID(geom)) As lat from souvenir where id='$info'";
      //Menampilkan data souvenir untuk detail informasi
$hasil=pg_query($querysearch);
while($row = pg_fetch_array($hasil)) //Perulangan menampilkan data
      {
              $id=$row['id'];
              $name=$row['name'];
              $address=$row['address'];
              $cp=$row['cp'];
              $owner=$row['owner'];
              $longitude=$row['lng'];
              $latitude=$row['lat'];

$dataarray[]=array('id'=>$id,'name'=>$name,'address'=>$address,'owner'=>$
owner,'cp'=>$cp,'longitude'=>$longitude,'latitude'=>$latitude);
      }
echo json_encode ($dataarray);
?>
```

c. Contoh luaran program



Informasi detail pariwisata yang dipilih

PROGRAM APLIKASI GIS MONITORING TANAH DAN KEPENDUDUKAN NAGARI PADANG LUA

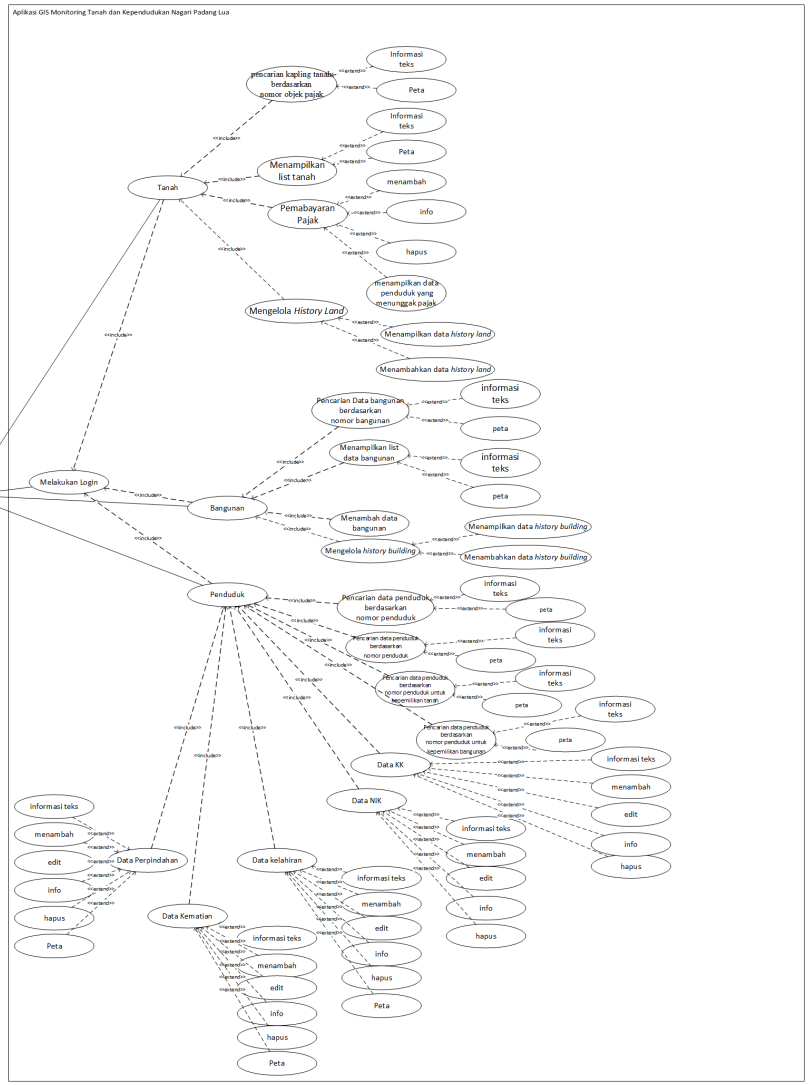
SURYA AFNARIUS DAN ANGGI TAMARA
SISTEM INFORMASI
UNIVERSITAS ANDALAS

PROGRAM APLIKASI GIS MONITORING TANAH DAN KEPENDUDUKAN NAGARI PADANG LUA

Ada 12 (Dua Belas) modul program/ fitur/ fungsional, yaitu modul program untuk :

1. Pencarian kapling tanah berdasarkan nomor objek pajak
2. Menampikan list tanah
3. Mengelola data land history
4. Pencarian data bangunan berdasarkan nomor bangunan
5. Menampikan list bangunan
6. Mengelola data building history
7. Pencarian kepemilikan tanah berdasarkan nomor penduduk
8. Pencarian data kepemilikan bangunan berdasarkan nomor penduduk
9. Pencarian data penduduk berdasarkan nomor penduduk
10. Mengelola data kelahiran
11. Mengelola data kematian
12. Mengelola data perpindahan

Usecase diagram Aplikasi GIS Monitoring Tanah dan Kependudukan Nagari Padang Lua



Dua Belas modul program:

1. Pencarian Kapling Tanah Berdasarkan Nomor Objek Pajak

a. Use Case Scenario

Tabel 1. Prosedur Pencarian Kapling Tanah Berupa Informasi Teks

Use case name	Pencarian kapling tanah berupa informasi teks
Participating Actor	*User*
Flow Of Events	1. *User* memilih menu *land*
	2. Sistem akan menampilkan sub menu
	3. *User* memilih menu *land search*
	4. *Sistem* akan menampilkan menu *land search*
	2. *User* memasukan nomor objek pajak
	3. Sistem akan menampilkan informasi teks mengenai kapling tanah
Entry condition	*User* telah login
Exit Condition	*User* dapat melihat informasi teks mengenai kapling tanah yang diinginkan
Alternative Course	1. *User* memilih menu *land*
	2. Sistem akan menampilkan sub menu
	3. *User* memilih menu *land search*
	4. *Sistem* akan menampilkan menu *land search*
	4. *User* memasukan nomor objek pajak
	5. Sistem akan menampilkan "*Data Not Found'*

Tabel 2. Prosedur Pencarian Kapling Tanah Berupa Peta

Use case name	Pencarian kapling tanah berupa peta
Participating Actor	*User*
Flow Of Events	1. *User* memilih menu *land*
	2. Sistem akan menampilkan sub menu
	3. *User* memilih menu *land search*
	4. Sistem akan menampilkan menu *land search*
	5. *User* memasukan nomor objek pajak
	6. Sistem akan menampilkan data objek pajak
	7. *User* memilih tab *map*
	6. Sistem akan menampilkan lokasi berdasarkan nomor objek pajak yang diinputkan
Entry condition	*User* telah login
Exit Condition	*User* dapat melihat lokasi kapling tanah

b. Kode Program

```
// Pencarian Kapling Tanah Berdasarkan No Objek Pajak Berupa Informasi Teks
// Method untuk memanggil data dari database menggunakan bahasa
// pemrograman php
public function find_object($id){
// query berdasarkan pencarian tanah
$sql = "SELECT la.nop, la.high_property, la.pawning,
la.tax_value, la.land_size,
la.building_size,la.id_building,p.pemilik_id,c.name as
pemilikrumah,lp.NIK , jo.name AS address
FROM land AS la
left join \"pemilikRumah\" p ON p.id_building=la.id_building
left join land_proprietary lp on lp.nop=la.nop
left join citizen c on (c.NIK=p.pemilik_id) , jorong AS jo
WHERE ST_CONTAINS(jo.geom, ST_CENTROID(la.geom)) AND la.nop =
'$id'";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
//generate hasil Query (informasi dari tanah)
$result = pg_fetch_assoc($Query);
$data['nop'] = $result['nop'];
$data['high_property'] = $result['high_property'];
$data['pawning'] = $result['pawning'];
$data['tax_value'] = $result['tax_value'];
$data['land_size'] = $result['land_size'];
$data['building_size'] = $result['building_size'];
$data['address'] = $result['address'];
$data['owner_rumah'] = $result['pemilikrumah'];
```

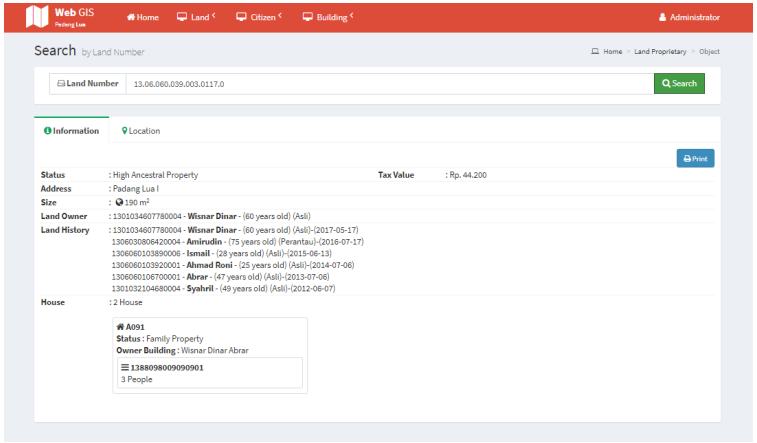
```
//Query memilih pemilik tanah

$sql = "SELECT ci.NIK, ci.name AS name, cs.name AS status,
EXTRACT(YEAR FROM AGE(ci.born_date)) AS age FROM land_proprietary
As lp LEFT JOIN citizen AS ci ON lp.NIK = ci.NIK LEFT JOIN
citizen_status AS cs ON ci.status_id = cs.status_id WHERE lp.nop
= '".$result['nop']."'";
$Query_owner = pg_Query($sql);
$data_owner = null;
if(pg_num_rows($Query_owner)>0){
$data_owner = array();
while ($result_sub = pg_fetch_assoc($Query_owner)) {
$data_owner[] = $result_sub;
}}
$data['owner'] = $data_owner;);
}}
$data['object'] = $data_object;
return $data;
}
else{
return 0;}}
```

//Kode Program Pencarian Kapling Tanah Berdasarkan No Objek Pajak Berupa

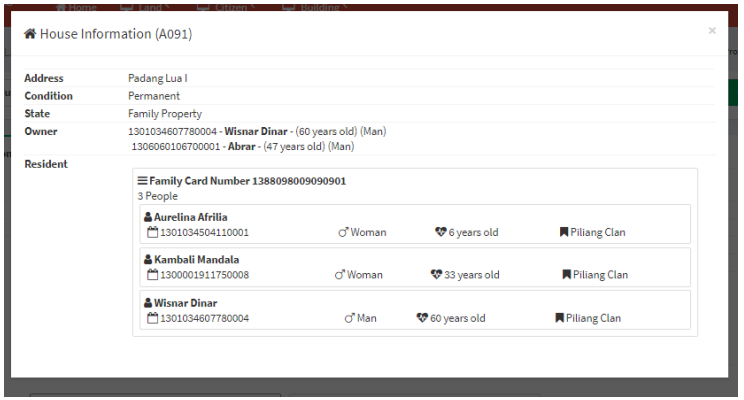
// Peta

```
// method mendapatkan layer dari object tanah berdasarkan nomor
// object tanah
public function object_layer($object_id){
$sql  = "SELECT ST_AsGeoJSON(la.geom) As geometry, la.nop,
ST_Y(ST_CENTROID(la.geom)) AS lat, ST_X(ST_CENTROID(la.geom)) AS
lng FROM land As la WHERE la.nop = '$object_id'";
$geojson = array(
'type'      => 'FeatureCollection',
'features'  => array()
);
$Query = pg_Query($sql);
if(pg_num_rows($Query)==0) return 0;
while($edge=pg_fetch_assoc($Query)){
$feature = array(
"type" => 'Feature',
'geometry' => json_decode($edge['geometry'], true),
'properties' => array(
'object' => $edge['nop'],
'center' => array(
'lat' => $edge['lat'],
'lng' => $edge['lng']
)));
array_push($geojson['features'], $feature);
  }
return $geojson;
}
```

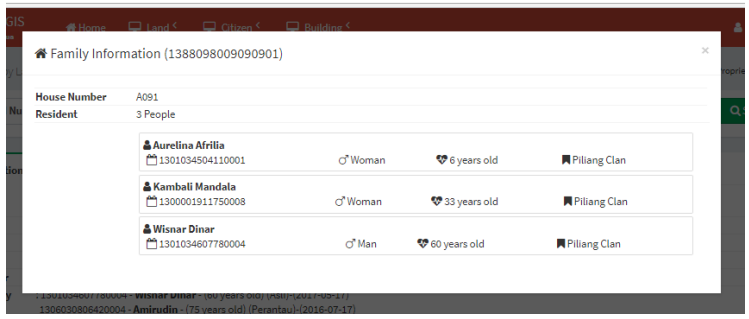
c. Contoh Luaran Program



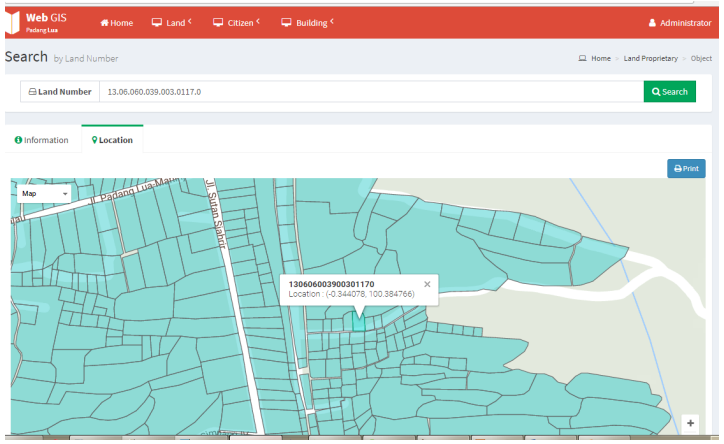
Gambar 1. Hasil Data Normal Mencari Kapling tanah Berupa Informasi Teks



Gambar 2. House Information



Gambar 3. *Family Information*



Gambar 4. Hasil Pencarian Kapling Tanah Berupa Peta

2. Menampikan *List* Tanah

a. Use Case Scenario

Tabel 3. Skenario Menampikan List Tanah Berupa Informasi Teks

Use Case Name	Menampikan *Land List* Berupa Informasi Teks
Participating Actor	*User*
Flow Of Events	1. *User* memilih menu *land*
	2. Sistem akan menampilkan submenu
	3. *User* memilih menu *land list*
	4. Sistem akan menampilkan menu *land list*
	5. *User* memilih keriteria yang telah disediakan dan mengklik *button show*
	6. Sistem menampilkan data berdasarkan kriteria yang telah dipilih
Entry condition	*User* telah login
Exit Condition	Sistem Menampilkan data bedasarkan kriteria yang dipilih

Tabel 4. Prosedur Menampikan List Tanah Berupa Peta

Use Case Name	Menampikan Land List Berupa Peta
Participating Actor	*User*
Flow Of Events	1. *User* memilih menu *land*
	2. Sistem akan menampilkan submenu
	3. *User* memilih menu *land list*
	4. Sistem akan menampilkan menu *land list*
	5. *User* memilih tab *map*
	6. Sistem menampilkan halaman tab *map*
	7. *User* memilih keriteria yang telah disediakan dan mengklik *button show*
	8. Sistem menampilkan peta berdasarkan kriteria yang telah dipilih
Entry condition	*User* telah login
Exit Condition	Sistem Menampilkan data bedasarkan kriteria yang dipilih

b. Kode Program

```
// Menampilkan List Tanah Berupa Informasi Teks
//method untuk mendapatkan daftar object tanah;
public function object_list($clan,$status,$owned,$pawned)
{
$condition_owner = array();
$condition = array();
$where_owner = "";
$where = "";
$where_end = "";
if($clan!="all"){
$cond = " MD5(ci.clan_id::teks)= '$clan'";
array_push($condition_owner, $cond);
$where_end = ' WHERE owner IS NOT NULL';
}
if($status!="all"){
$cond = 'la.high_property = '.$status;
array_push($condition, $cond);
}
if($owned!="all"){
$cond = 'ci.status_id = '.$owned;
array_push($condition_owner, $cond);
$where_end = ' WHERE owner IS NOT NULL';
}
if($pawned!="all"){
$cond = 'la.pawning = '.$pawned;
array_push($condition, $cond);
}
array_push($condition_owner, "lp.nop = la.nop");
array_push($condition, "true");
$where_owner = "WHERE " . implode(" AND ", $condition_owner);
$where = "WHERE " . implode(" AND ", $condition);
```

```
$sql = "SELECT * FROM (SELECT la.nop, la.land_size,
ci.building_size,(SELECT array_to_json(array_agg(f)) FROM (SELECT
ci.nik, ci.name AS name, cs.name AS status, EXTRACT(YEAR FROM
AGE(ci.born_date)) AS age FROM land_proprietary As lp LEFT JOIN
pemilik AS ci ON lp.nik = ci.nik LEFT JOIN citizen_status AS cs ON
ci.status_id = cs.status_id $where_owner) AS f) AS owner FROM land
AS la left join land_proprietary As lp on la.nop=lp.nop LEFT JOIN
pemilik AS ci ON lp.nik = ci.nik $where) as fs $where_end";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data['nop'] = $data['nop'];
$data['land_size'] = $data['land_size'];
$data['building_size'] = $data['building_size'];
$data['owner'] = json_decode($data['owner']);
$data_a[] = $data;
}
return $data_a;
}
else{
return 0;
}}

// Menampilkan List Tanah Berupa Peta
//file proses.php
else if($do == 'g_map_land_clan'){
header('Content-Type: application/json');
$current_selected_clan= $_GET['current_selected_clan'];
$current_selected_status=$_GET['current_selected_status'];
$current_selected_owner=$_GET['current_selected_owner'];
$current_selected_pawned =$_GET['current_selected_pawned'];
if($current_selected_clan == 'all' && $current_selected_status ==
'all' && $current_selected_owner == 'all' &&
$current_selected_pawned == 'all'){
$m = new land();
$data = $m->load_land_map();
echo json_encode($data);
}
else if($current_selected_clan == 'all' && $current_selected_status
!= 'all' && $current_selected_owner == 'all' &&
$current_selected_pawned != 'all'){
$m = new land();
$data = $m->load_land_map1
($current_selected_status,$current_selected_pawned);
echo json_encode($data);
}

else if($current_selected_clan == 'all' && $current_selected_status
!= 'all' && $current_selected_owner == 'all' &&
$current_selected_pawned == 'all'){
$m = new land();
$data = $m->load_land_map2($current_selected_status );
echo json_encode($data);
}
```

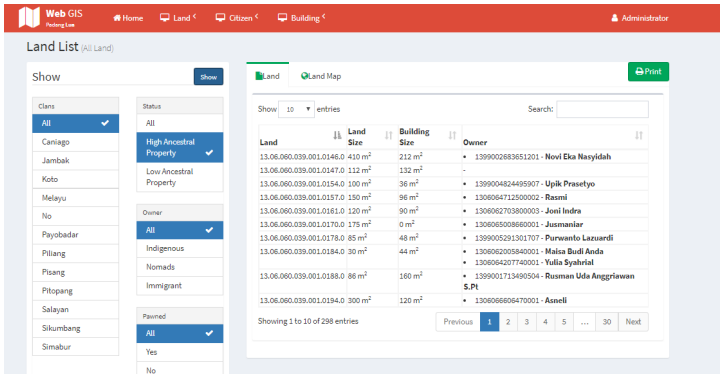
```
else if($current_selected_clan == 'all' && $current_selected_status
== 'all' && $current_selected_owner != 'all' &&
$current_selected_pawned == 'all'){
$m = new land();
$data = $m->load_land_map3($current_selected_owner );
echo json_encode($data);
}
else if($current_selected_clan == 'all' && $current_selected_status
== 'all' && $current_selected_owner == 'all' &&
$current_selected_pawned != 'all'){
$m = new land();
$data = $m->load_land_map4($current_selected_pawned);
echo json_encode($data);
}
else if($current_selected_clan == 'all' && $current_selected_status
== 'all' && $current_selected_owner != 'all' &&
$current_selected_pawned != 'all'){
$m = new land();
$data = $m-
>load_land_map5($current_selected_owner,$current_selected_pawned);e
cho json_encode($data);
}
else if(strlen($current_selected_clan) == 32 &&
$current_selected_status == 'all' && $current_selected_owner ==
'all' && $current_selected_pawned == 'all'){
$m = new land();
$data = $m->load_land_map6($current_selected_clan);
echo json_encode($data);
}
else if(strlen($current_selected_clan) == 32 &&
$current_selected_status != 'all' && $current_selected_owner ==
'all' && $current_selected_pawned == 'all'){
$m = new land();
$data = $m-
>load_land_map7($current_selected_clan,$current_selected_status);
echo json_encode($data);
}
else if(strlen($current_selected_clan) == 32 &&
$current_selected_status != 'all' && $current_selected_owner !=
'all' && $current_selected_pawned == 'all'){
$m = new land();
$data = $m-
>load_land_map12($current_selected_clan,$current_selected_status,$c
urrent_selected_owner);
echo json_encode($data);
}
else if(strlen($current_selected_clan) == 32 &&
$current_selected_status == 'all' && $current_selected_owner !=
'all' && $current_selected_pawned == 'all'){
$m = new land();
$data = $m-
>load_land_map8($current_selected_clan,$current_selected_owner);
echo json_encode($data);
}
```

```
else if(strlen($current_selected_clan) == 32 &&
$current_selected_status == 'all' && $current_selected_owner ==
'all' && $current_selected_pawned != 'all'){
$m = new land();
$data = $m-
>load_land_map9($current_selected_clan,$current_selected_pawned);
echo json_encode($data);
}
else if(strlen($current_selected_clan) == 32 &&
$current_selected_status == 'all' && $current_selected_owner !=
'all' && $current_selected_pawned != 'all'){
$m = new land();
$data = $m-
>load_land_map10($current_selected_clan,$current_selected_owner,cur
rent_selected_pawned);
echo json_encode($data);
}
else if(strlen($current_selected_clan) == 32 &&
$current_selected_status != 'all' && $current_selected_owner !=
'all' && $current_selected_pawned != 'all'){
$m = new land();
$data = $m-
>load_land_map11($current_selected_clan,$current_selected_status,$c
urrent_selected_owner,$current_selected_pawned );
echo json_encode($data);
}
else{
echo 0;
}}
 ---------------------------------------------------------------
// file Class.php ( method mendapatkan layer dari list tanah)
public function load_land_map(){
$sql="SELECT ST_AsGeoJSON(la.geom) As geometry,
ST_Y(ST_CENTROID(la.geom)) AS lat, ST_X(ST_CENTROID(la.geom)) AS
lng,  la.nop, case when MD5(ci.clan_id::teks)='all' then 1 else 0
end as a FROM land as la LEFT JOIN  land_proprietary As lp on
la.nop= lp.nop LEFT JOIN citizen AS ci ON lp.nik = ci.nik LEFT JOIN
citizen_status AS cs ON ci.status_id = cs.status_id";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;
}
return $data_a;
}}

public function
load_land_map1($current_selected_status,$current_selected_pawned){$
sql="SELECT ST_AsGeoJSON(la.geom) As geometry,
ST_Y(ST_CENTROID(la.geom)) AS lat, ST_X(ST_CENTROID(la.geom)) AS
lng,  la.nop, case when la.high_property='$current_selected_status'
AND la.pawning='$current_selected_pawned' then 1 else 0 end as a
FROM land as la LEFT JOIN  land_proprietary As lp on la.nop= lp.nop
LEFT JOIN citizen AS ci ON lp.nik = ci.nik LEFT JOIN citizen_status
AS cs ON ci.status_id = cs.status_id";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;
}
return $data_a;
}}
public function load_land_map2 ($current_selected_status){
$sql="SELECT ST_AsGeoJSON(la.geom) As geometry,
ST_Y(ST_CENTROID(la.geom)) AS lat, ST_X(ST_CENTROID(la.geom)) AS
lng,  la.nop, case when la.high_property='$current_selected_status'
then 1 else 0 end as a FROM land as la LEFT JOIN  land_proprietary
As lp on la.nop= lp.nop LEFT JOIN citizen AS ci ON lp.nik = ci.nik
LEFT JOIN citizen_status AS cs ON ci.status_id = cs.status_id";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;
}
return $data_a;
}}
public function load_land_map4 ($current_selected_pawned){
$sql="SELECT ST_AsGeoJSON(la.geom) As geometry,
ST_Y(ST_CENTROID(la.geom)) AS lat, ST_X(ST_CENTROID(la.geom)) AS
lng,  la.nop, case when la.pawning='$current_selected_pawned' then
1 else 0 end as a FROM land as la LEFT JOIN  land_proprietary As lp
on la.nop= lp.nop LEFT JOIN citizen AS ci ON lp.nik = ci.nik LEFT
JOIN citizen_status AS cs ON ci.status_id = cs.status_id";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;
}
return $data_a;
}}
public function load_land_map5
($current_selected_owner,$current_selected_pawned){
$sql="SELECT ST_AsGeoJSON(la.geom) As geometry,
ST_Y(ST_CENTROID(la.geom)) AS lat, ST_X(ST_CENTROID(la.geom)) AS
lng,  la.nop, case when ci.status_id='$current_selected_owner' AND
la.pawning='$current_selected_pawned' then 1 else 0 end as a FROM
land as la LEFT JOIN  land_proprietary As lp on la.nop= lp.nop LEFT
JOIN citizen AS ci ON lp.nik = ci.nik LEFT JOIN citizen_status AS
cs ON ci.status_id = cs.status_id";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;
}
return $data_a;
}}

public function load_land_map6 ($current_selected_clan){
$sql="SELECT ST_AsGeoJSON(la.geom) As geometry,
ST_Y(ST_CENTROID(la.geom)) AS lat, ST_X(ST_CENTROID(la.geom)) AS
```

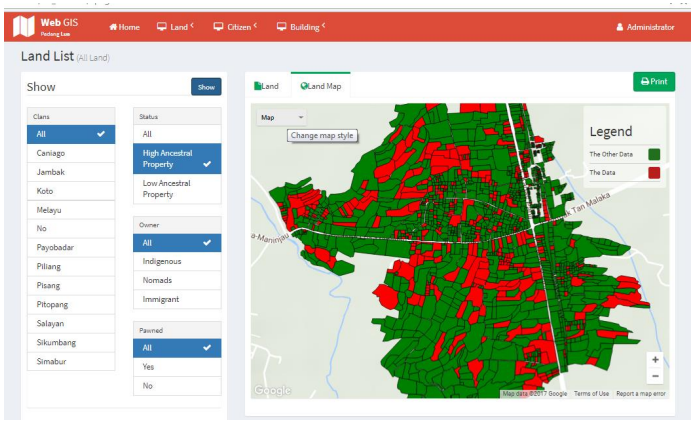
```
lng,  la.nop, case when
MD5(ci.clan_id::teks)='$current_selected_clan' then 1 else 0 end as
a FROM land as la LEFT JOIN  land_proprietary As lp on la.nop=
lp.nop LEFT JOIN citizen AS ci ON lp.nik = ci.nik LEFT JOIN
citizen_status AS cs ON ci.status_id = cs.status_id";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;
}
return $data_a;
}}
public function load_land_map7 ($current_selected_clan
,$current_selected_status){
$sql="SELECT ST_AsGeoJSON(la.geom) As geometry,
ST_Y(ST_CENTROID(la.geom)) AS lat, ST_X(ST_CENTROID(la.geom)) AS
lng,  la.nop, case when
MD5(ci.clan_id::teks)='$current_selected_clan' AND
la.high_property='$current_selected_status' then 1 else 0 end as a
FROM land as la LEFT JOIN  land_proprietary As lp on la.nop= lp.nop
LEFT JOIN citizen AS ci ON lp.nik = ci.nik LEFT JOIN citizen_status
AS cs ON ci.status_id = cs.status_id";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
 $data_a[] = $data;
}
return $data_a;
}}
public function load_land_map8
($current_selected_clan,$current_selected_owner){
$sql="SELECT ST_AsGeoJSON(la.geom) As geometry,
ST_Y(ST_CENTROID(la.geom)) AS lat, ST_X(ST_CENTROID(la.geom)) AS
lng,  la.nop, case when
MD5(ci.clan_id::teks)='$current_selected_clan' AND
ci.status_id='$current_selected_owner' then 1 else 0 end as a FROM
land as la LEFT JOIN  land_proprietary As lp on la.nop= lp.nop LEFT
JOIN citizen AS ci ON lp.nik = ci.nik LEFT JOIN citizen_status AS
cs ON ci.status_id = cs.status_id";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;
}
return $data_a;
 }}
public function load_land_map9
($current_selected_clan,$current_selected_pawned){
$sql="SELECT ST_AsGeoJSON(la.geom) As geometry,
ST_Y(ST_CENTROID(la.geom)) AS lat, ST_X(ST_CENTROID(la.geom)) AS
lng,  la.nop, case when
MD5(ci.clan_id::teks)='$current_selected_clan' AND
la.pawning='$current_selected_pawned' then 1 else 0 end as a FROM
land as la LEFT JOIN  land_proprietary As lp on la.nop= lp.nop LEFT
JOIN citizen AS ci ON lp.nik = ci.nik LEFT JOIN citizen_status AS
cs ON ci.status_id = cs.status_id";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;
}
return $data_a;
}}
public function load_land_map10
($current_selected_clan,$current_selected_owner,$current_selected_p
awned){
$sql="SELECT ST_AsGeoJSON(la.geom) As geometry,
ST_Y(ST_CENTROID(la.geom)) AS lat, ST_X(ST_CENTROID(la.geom)) AS
lng,  la.nop, case when
MD5(ci.clan_id::teks)='$current_selected_clan' AND
ci.status_id='$current_selected_owner' AND
la.pawning='$current_selected_pawned' then 1 else 0 end as a FROM
land as la LEFT JOIN  land_proprietary As lp on la.nop= lp.nop LEFT
JOIN citizen AS ci ON lp.nik = ci.nik LEFT JOIN citizen_status AS
cs ON ci.status_id = cs.status_id";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;
}
return $data_a;
}}
public function load_land_map11
($current_selected_clan,$current_selected_status,$current_selected_
owner,$current_selected_pawned){
$sql="SELECT ST_AsGeoJSON(la.geom) As geometry,
ST_Y(ST_CENTROID(la.geom)) AS lat, ST_X(ST_CENTROID(la.geom)) AS
lng,  la.nop, case when
MD5(ci.clan_id::teks)='$current_selected_clan' AND
la.high_property='$current_selected_status' AND
ci.status_id='$current_selected_owner' AND
la.pawning='$current_selected_pawned' then 1 else 0 end as a FROM
land as la LEFT JOIN  land_proprietary As lp on la.nop= lp.nop LEFT
JOIN citizen AS ci ON lp.nik = ci.nik LEFT JOIN citizen_status AS
cs ON ci.status_id = cs.status_id";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;}
return $data_a;
}}
public function load_land_map12
($current_selected_clan,$current_selected_status,$current_selected_
owner){
$sql="SELECT ST_AsGeoJSON(la.geom) As geometry,
ST_Y(ST_CENTROID(la.geom)) AS lat, ST_X(ST_CENTROID(la.geom)) AS
lng,  la.nop, case when
MD5(ci.clan_id::teks)='$current_selected_clan' AND
la.high_property='$current_selected_status' AND
```

```
ci.status_id='$current_selected_owner'  then 1 else 0 end as a FROM
land as la LEFT JOIN  land_proprietary As lp on la.nop= lp.nop LEFT
JOIN citizen AS ci ON lp.nik = ci.nik LEFT JOIN citizen_status AS
cs ON ci.status_id = cs.status_id";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
 $data_a[] = $data;
}
return $data_a;
}}
public function load_land_map3 ($current_selected_owner){
 $sql="SELECT ST_AsGeoJSON(la.geom) As geometry,
ST_Y(ST_CENTROID(la.geom)) AS lat, ST_X(ST_CENTROID(la.geom)) AS
lng,  la.nop, case when ci.status_id='$current_selected_owner' then
1 else 0 end as a FROM land as la LEFT JOIN  land_proprietary As lp
on la.nop= lp.nop LEFT JOIN citizen AS ci ON lp.nik = ci.nik LEFT
JOIN citizen_status AS cs ON ci.status_id = cs.status_id";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;
}
return $data_a;}}
```

c. Contoh luaran program



Gambar 5. Hasil Menampilkan *List* Tanah Berupa Informasi Teks



Gambar 6. Hasil Menampilkan *List* Tanah Berupa Peta

3. Mengelola Data *Land History*

a. Use Case Skenario

Tabel 5. Skenario Menampilkan Data Land history

Use Case Name	Menampilkan data History Land
Participating Actor	*User*
Flow Of Events	1. *User* memilih menu *land*
	2. Sistem akan menampilkan sub menu
	3. *User* memilih menu *land history*
	4. Sistem akan menampilkan data *history land*
Entry condition	*User* telah login
Exit Condition	*User* dapat melihat informasi *history land*

Tabel 6. Skenario Menginputkan Data Land history

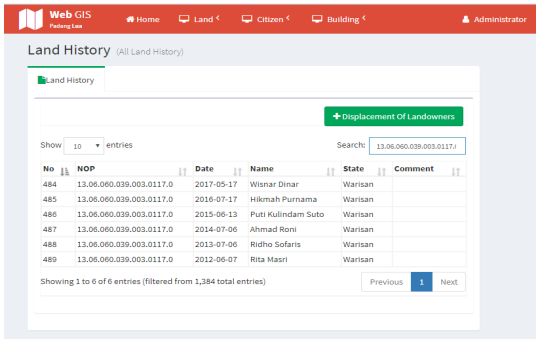
Use Case Name	Menambah Data History Land
Participating Actor	*User*
Flow Of Events	*1. User* memilih menu *land*
	2. Sistem akan menampilkan sub menu
	3. User memilih menu *land history*

	4. Sistem akan menampilkan menu *land history*
	5. *User* mengklik *buttton displacement of land owner*
	6. Sistem akan form menambah data *history land*
	7. *User* mengisi *form* dan memilih *button save*
	8. Sistem akan menyimpan perubahan dalam *database*
Entry condition	*User* telah login
Exit Condition	Sistem menyimpan perubahan data ke *database* dan membuka kembali halaman *land history*

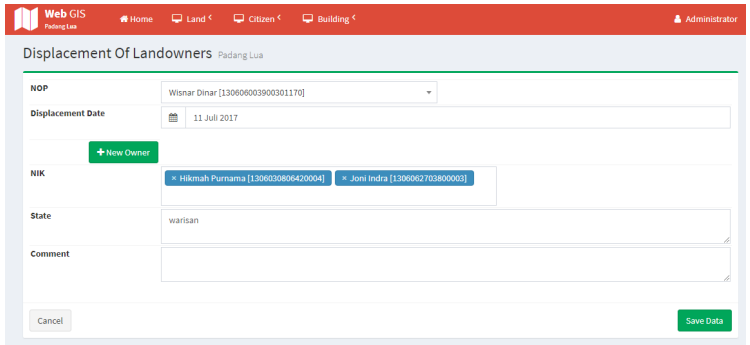
b. Kode Program

```
//Kode Program Menampilkan Data History Land
//method untuk query menampilkan data history land
public function load_history(){
$sql="select b.nop, b.state, b.tgl, c.name, b.coment FROM
land_proprietary as b  LEFT JOIN pemilik as c ON b.nik=c.nik  ORDER
BY  nop, tgl desc ";
$query = pg_query($sql);
if(pg_num_rows($query) > 0){
while ($data = pg_fetch_assoc($query)){
$data_a[] = $data;
}
return $data_a;
}}
// Menginputkan Data History Land
//method untuk query menginputkan data history land
public function insert($data){
$sqlICN="INSERT INTO land_proprietary(nop, nik, state, tgl, coment)
VALUES ";
$sqlICN.="(".$data['nop'].",'".$data['nik']."','".$data['state
']."','".$data['tgl']."','".$data['coment']."')";
$sql=$sqlICN.';';
$query = pg_query($sql);
if($query){
return 1;
}
else{
return 0;
}
```

c. Contoh Luaran Program



Gambar 4. Hasil Menampilkan Data Land history



Gambar 5. Hasil Menginputkan Data Land history

4. Pencarian Data Bangunan Berdasarkan Nomor Bangunan

a. Use Case Skenario

Tabel 7. Sknario Pencarian Data Bangunan Berdasarkan Nomor Bangunan Berupa Informasi teks

Use case name	Pencarian Data Bangunan Berdasarkan Nomor Bangunan Berupa Informasi teks
Participating Actor	*User*
Flow Of Events	1. *User* memilih menu *building*
	2. Sistem akan menampilkan sub menu
	3. *User* memilih menu *building search*

	4. *Sistem* akan menampilkan menu *building search*
	5. *User* memasukan nomor bangunan
	6. Sistem akan menampilkan informasi teks mengenai kapling bangunan
Entry condition	*User* telah login
Exit Condition	*User* dapat melihat informasi teks mengenai kapling tanah yang diinginkan
Alternative Course	a. *User* memilih menu *building*
	b. Sistem akan menampilkan sub menu
	c. *User* memilih menu *building search*
	d. *Sistem* akan menampilkan menu *building search*
	7. *User* memasukan nomor bangunan
	8. Sistem akan menampilkan "*Data Not Found'*

Tabel 8. Skenario Pencarian Data Bangunan Berdasarkan Nomor Bangunan Berupa Peta

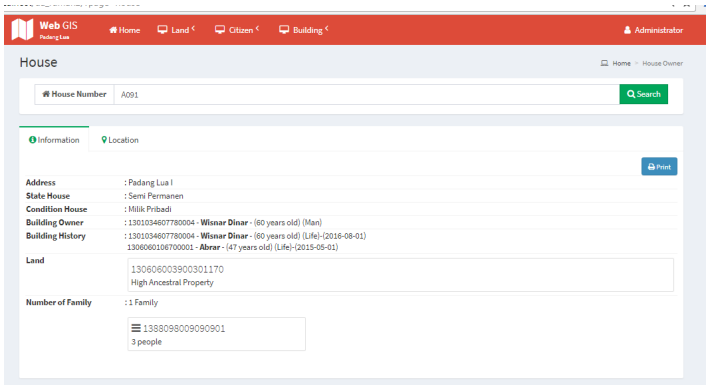
Use case name	Pencarian Data Bangunan Berdasarkan Nomor Bangunan Berupa Peta
Participating Actor	*User*
Flow Of Events	1. *User* memilih menu *building*
	2. Sistem akan menampilkan sub menu
	3. *User* memilih menu *building search*
	4. Sistem akan menampilkan menu *building search*
	5. *User* memasukan nomor bangunan
	6. Sistem akan menampilkan informasi teks mengenai kapling bangunan
	7. *User* memilih tab *map*
	8. Sistem akan menampilkan lokasi berdasarkan nomor bangunan yang diinputkan
Entry condition	*User* telah login
Exit Condition	*User* dapat melihat lokasi bangunan berdasarkan nomor bangunan

b. Kode Program

```
// Pencarian Data Bangunan Berdasarkan Nomor Bangunan Berupa Informasi
// Teks
// Method untuk memanggil data dari database menggunakan bahasa
// pemrograman php
public function find_house($id){
// query berdasarkan pencarian bangunan
$sql = "SELECT ho.no_house, ho.status, ho.condition,la.nop AS
nop, la.high_property AS land_status, jo.name AS address FROM
building AS ho, land AS la, jorong AS jo WHERE
ST_CONTAINS(la.geom, ST_CENTROID(ho.geom)) AND
ST_CONTAINS(jo.geom, ST_CENTROID(la.geom)) AND
ST_CONTAINS(jo.geom, ST_CENTROID(ho.geom)) AND ho.no_house =
'$id'";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
//generate informasi bangunan;
$result = pg_fetch_assoc($Query);
$data['no_house'] = $result['no_house'];
$data['status'] = $result['status'];
$data['condition'] = $result['condition'];
$data['land']['nop'] = $result['nop'];
$data['land']['status'] = $result['land_status'];
$data['address'] = $result['address'];
//mencari pemilik rumah
$sql = "SELECT ci.nik, ci.name, ci.gender, EXTRACT(YEAR FROM
AGE(ci.born_date)) AS age, ci.status FROM building as b LEFT JOIN
building_family AS how ON b.id=how.id_building LEFT JOIN citizen
AS ci ON how.family_no = ci.family_no WHERE b.no_house  =
'".$result['no_house']."'";
$Query_owner = pg_Query($sql);
$data_owner = null;
if(pg_num_rows($Query_owner)>0){
$data_owner = array();
while ($result_sub = pg_fetch_assoc($Query_owner)) {
$data_owner[] = $result_sub;
} }
$data['owner'] = $data_owner;
```

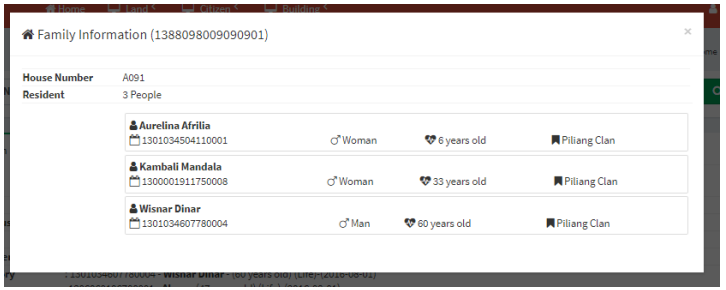
```
//mencari penghuni rumah
$sql = "SELECT ci.family_no, COUNT(ci.nik) AS resident
FROM citizen AS ci
LEFT JOIN building_family as b ON ci.family_no=b.family_no
LEFT JOIN building as c ON b.id_building=c.id
WHERE no_house = '".$result['no_house']."' GROUP BY
ci.family_no";
$Query_fcn = pg_Query($sql);
$data_fcn = null;
if(pg_num_rows($Query_fcn)>0){
$data_fcn = array();
while ($result_sub_fcn = pg_fetch_assoc($Query_fcn)) {
$data_fcn[] = $result_sub_fcn;
}}
$data['resident'] = $data_fcn;
return $data;
}else{
return 0;
}}
// Pencarian Data Bangunan Berdasarkan Nomor Bangunan Berupa Peta
//method mendapatkan layer dari object bangunan berdasarkan nomor
bangunan
public function house_layer($no_house){
$sql  = "SELECT ST_AsGeoJSON(ho.geom) As geometry, ho.no_house,
ST_Y(ST_CENTROID(ho.geom)) AS lat, ST_X(ST_CENTROID(ho.geom)) AS
lng FROM building As ho WHERE ho.no_house = '$no_house'";
$geojson = array(
'type'      => 'FeatureCollection',
'features'  => array()
);
$Query = pg_Query($sql);
if(pg_num_rows($Query)==0)
return 0;
while($edge=pg_fetch_assoc($Query)){
$feature = array(
"type" => 'Feature',
'geometry' => json_decode($edge['geometry'], true),
'properties' => array(
'object' => $edge['no_house'],
'center' => array(
'lat' => $edge['lat'],
'lng' => $edge['lng']
))
);
array_push($geojson['features'], $feature);
}
return $geojson;
}
```

c. Contoh Luaran Program

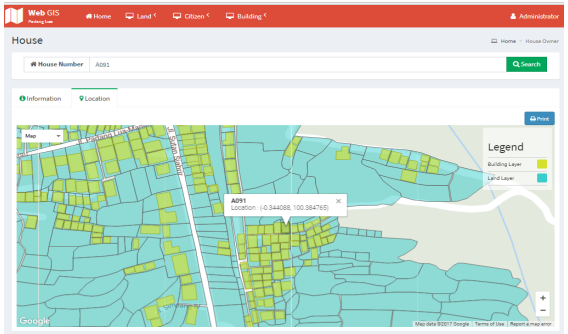


Gambar 9. Hasil Data Normal Mencari Bangunan Berdasarkan Nomor Bangunan

Gambar 10. *Land Information*



Gambar 11. *Family Information*



Gambar 12. Hasil Mencari Lokasi Bangunan Berdasarkan Nomor Bangunan

5. Menampikan *List* Bangunan

a. Use Case Skenario

Tabel 9. Skenario Menampikan *List* Bangunan Berupa Informasi Teks

Use Case Name	Menampikan *Building List* Berupa Informasi Teks
Participating Actor	*User*
Flow Of Events	1. *User* memilih menu *building*
	2. Sistem akan menampilkan submenu
	3. *User* memilih menu *building list*
	4. Sistem akan menampilkan menu *building list*
	5. *User* memilih keriteria yang telah disediakan dan mengklik *button show*
	6. Sistem menampilkan data berdasarkan kriteria yang telah dipilih
Entry condition	*User* telah login
Exit Condition	Sistem Menampilkan data bedasarkan kriteria yang dipilih

Tabel 10. Skenario Menampikan *List* Bangunan Berupa Peta

Use Case Name	Menampikan *building* List Berupa Peta
Participating Actor	*User*
Flow Of Events	1. *User* memilih menu *building*
	2. Sistem akan menampilkan submenu
	3. *User* memilih menu *buildng list*
	4. Sistem akan menampilkan menu *building list*
	5. *User* memilih tab *map*
	6. Sistem menampilkan halaman tab *map*
	7. *User* memilih keriteria yang telah disediakan dan mengklik *button show*
	8. Sistem menampilkan peta berdasarkan kriteria yang telah dipilih
Entry condition	*User* telah login
Exit Condition	Sistem Menampilkan data bedasarkan kriteria yang dipilih

b. Kode Program

```
//Menampilkan List Bangunan Berupa Informasi Teks
// file class.php
else if($do == 'g_building_all'){
header('Content-Type: application/json');
$current_selected_clan=$_GET['current_selected_clan'];
$current_selected_owner=$_GET['current_selected_owner'];
$current_selected_owner_land =$_GET['current_selected_owner_land'];
$current_selected_status=$_GET['current_selected_status'];
if($current_selected_clan == 'all' && $current_selected_owner ==
'all' && $current_selected_owner_land == 'all' &&
$current_selected_status == 'all'){
$m = new building();
$data = $m->load();
echo json_encode($data);
}
else if($current_selected_clan == 'all' && $current_selected_owner
!= 'all' && $current_selected_owner_land != 'all' &&
$current_selected_status != 'all'){
$m = new building();
$data = $m->load_building_all1($current_selected_owner,
$current_selected_owner_land,$current_selected_status );
echo json_encode($data);
}
else if($current_selected_clan == 'all' && $current_selected_owner
== 'all' && $current_selected_owner_land != 'all' &&
$current_selected_status != 'all'){
$m = new building();
$data = $m-
>load_building_all5($current_selected_owner_land,$current_selected_
status );
echo json_encode($data);
}
else if($current_selected_clan == 'all' && $current_selected_owner
== 'all' && $current_selected_owner_land == 'all' &&
$current_selected_status != 'all'){
$m = new building();
$data = $m->load_building_all6($current_selected_status );
echo json_encode($data);
```

```
}
else if($current_selected_clan == 'all' && $current_selected_owner
!= 'all' && $current_selected_owner_land == 'all' &&
$current_selected_status != 'all'){
$m = new building();
$data = $m->load_building_all7($current_selected_owner,
$current_selected_status );
echo json_encode($data);
}
else if($current_selected_clan == 'all' && $current_selected_owner
!= 'all' && $current_selected_owner_land == 'all' &&
$current_selected_status == 'all'){
$m = new building();
$data = $m->load_building_all8($current_selected_owner);
echo json_encode($data);
}
else if(strlen($current_selected_clan) == 32 &&
$current_selected_owner == 'all' && $current_selected_owner_land !=
'all' && $current_selected_status != 'all'){
$m = new building();
$data = $m->load_building_all2($current_selected_clan,
$current_selected_owner_land,$current_selected_status );
echo json_encode($data);
}
else if(strlen($current_selected_clan) == 32 &&
$current_selected_owner != 'all' && $current_selected_owner_land ==
'all' && $current_selected_status != 'all'){
$m = new building();
$data = $m-
>load_building_all3($current_selected_clan,$current_selected_owner,
$current_selected_status );
echo json_encode($data);
}
else if(strlen($current_selected_clan) == 32 &&
$current_selected_owner != 'all' && $current_selected_owner_land !=
'all' && $current_selected_status == 'all'){
$m = new building();
$data = $m-
>load_building_all4($current_selected_clan,$current_selected_owner,
$current_selected_owner_land);
echo json_encode($data);
}
else if(strlen($current_selected_clan) == 32 &&
$current_selected_owner != 'all' && $current_selected_owner_land !=
'all' && $current_selected_status != 'all'){
$m = new building();
$data = $m-
>load_building_all($current_selected_clan,$current_selected_owner,
$current_selected_owner_land,$current_selected_status );
echo json_encode($data);
}
else{
echo 0;
}}
-------------------------------------------------------------
```

```
// file Class.php
public function
load_building_all1($current_selected_owner,$current_selected_owner_
land, $current_selected_status){
$sql="SELECT b.*,c.family_no,c.name  FROM building b LEFT JOIN
building_family f ON b.id=f.id_building LEFT JOIN citizen c ON
c.nik=b.nik LEFT JOIN clans as k on c.clan_id=k.clan_id LEFT JOIN
citizen_status as j on c.status_id=j.status_id LEFT JOIN
land_proprietary as m on m.nik=c.nik LEFT JOIN land as l ON
l.nop=m.nop LEFT JOIN owner_state as d ON d.id_state=l.id_state
where c.status_id='$current_selected_owner' AND
l.id_state='$current_selected_owner_land' AND
l.high_property='$current_selected_status'";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
$i=array();
while ($data = pg_fetch_assoc($Query)){
$data_a[$data['id']][] = $data;
}
return $data_a;
}}
public function load_building_all5($current_selected_owner_land,
$current_selected_status){
$sql="SELECT b.*,f.family_no,c.name  FROM building b LEFT JOIN
building_family f ON b.id=f.id_building LEFT JOIN citizen c ON
c.nik=b.nik LEFT JOIN clans as k on c.clan_id=k.clan_id LEFT JOIN
citizen_status as j on c.status_id=j.status_id LEFT JOIN
land_proprietary as m on m.nik=c.nik LEFT JOIN land as l ON
l.nop=m.nop LEFT JOIN owner_state as d ON d.id_state=l.id_state
where l.id_state='$current_selected_owner_land' AND
l.high_property='$current_selected_status'";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
$i=array();
while ($data = pg_fetch_assoc($Query)){
$data_a[$data['id']][] = $data;
}
return $data_a;
}}
public function load_building_all6($current_selected_status){
$sql="SELECT b.*,f.family_no,c.name  FROM building b LEFT JOIN
building_family f ON b.id=f.id_building LEFT JOIN citizen c ON
c.nik=b.nik LEFT JOIN clans as k on c.clan_id=k.clan_id LEFT JOIN
citizen_status as j on c.status_id=j.status_id LEFT JOIN
land_proprietary as m on m.nik=c.nik LEFT JOIN land as l ON
l.nop=m.nop LEFT JOIN owner_state as d ON d.id_state=l.id_state
where l.high_property='$current_selected_status'";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
$i=array();
while ($data = pg_fetch_assoc($Query)){
$data_a[$data['id']][] = $data;
}
return $data_a;
}
```

```
}

public function
load_building_all3($current_selected_clan,$current_selected_owner,$
current_selected_status){
$sql="SELECT b.*,f.family_no,c.name  FROM building b LEFT JOIN
building_family f ON b.id=f.id_building LEFT JOIN citizen c ON
c.nik=b.nik LEFT JOIN clans as k on c.clan_id=k.clan_id LEFT JOIN
citizen_status as j on c.status_id=j.status_id LEFT JOIN
land_proprietary as m on m.nik=c.nik LEFT JOIN land as l ON
l.nop=m.nop LEFT JOIN owner_state as d ON d.id_state=l.id_state
where MD5(c.clan_id::teks)='$current_selected_clan' AND
l.high_property='$current_selected_status'";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
$i=array();
while ($data = pg_fetch_assoc($Query)){
$data_a[$data['id']][] = $data;
}
return $data_a;
}}
public function
load_building_all7($current_selected_owner,$current_selected_status
){
$sql="SELECT b.*,f.family_no,c.name  FROM building b LEFT JOIN
building_family f ON b.id=f.id_building LEFT JOIN citizen c ON
c.nik=b.nik LEFT JOIN clans as k on c.clan_id=k.clan_id LEFT JOIN
citizen_status as j on c.status_id=j.status_id LEFT JOIN
land_proprietary as m on m.nik=c.nik LEFT JOIN land as l ON
l.nop=m.nop LEFT JOIN owner_state as d ON d.id_state=l.id_state
where c.status_id='$current_selected_owner' AND
l.high_property='$current_selected_status'";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
$i=array();
while ($data = pg_fetch_assoc($Query)){
$data_a[$data['id']][] = $data;
}
return $data_a;
}}
public function load_building_all8($current_selected_owner){
$sql="SELECT b.*,f.family_no,c.name  FROM building b LEFT JOIN
building_family f ON b.id=f.id_building LEFT JOIN citizen c ON
c.nik=b.nik LEFT JOIN clans as k on c.clan_id=k.clan_id LEFT JOIN
citizen_status as j on c.status_id=j.status_id LEFT JOIN
land_proprietary as m on m.nik=c.nik LEFT JOIN land as l ON
l.nop=m.nop LEFT JOIN owner_state as d ON d.id_state=l.id_state
where c.status_id='$current_selected_owner'";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
$i=array();
while ($data = pg_fetch_assoc($Query)){
$data_a[$data['id']][] = $data;
}
return $data_a;
```

```
}}

public function
load_building_all2($current_selected_clan,$current_selected_owner_l
and, $current_selected_status){
$sql="SELECT b.*,f.family_no,c.name  FROM building b LEFT JOIN
building_family f ON b.id=f.id_building LEFT JOIN citizen c ON
c.nik=b.nik LEFT JOIN clans as k on c.clan_id=k.clan_id LEFT JOIN
citizen_status as j on c.status_id=j.status_id LEFT JOIN
land_proprietary as m on m.nik=c.nik LEFT JOIN land as l ON
l.nop=m.nop LEFT JOIN owner_state as d ON d.id_state=l.id_state
where MD5(c.clan_id::teks)='$current_selected_clan' AND
l.id_state='$current_selected_owner_land' AND
l.high_property='$current_selected_status'";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
$i=array();
while ($data = pg_fetch_assoc($Query)){
$data_a[$data['id']][] = $data;
}
return $data_a;
}
}
public function
load_building_all($current_selected_clan,$current_selected_owner,$c
urrent_selected_owner_land, $current_selected_status){
$sql="SELECT b.*,f.family_no,c.name  FROM building b LEFT JOIN
building_family f ON b.id=f.id_building LEFT JOIN citizen c ON
c.nik=b.nik LEFT JOIN clans as k on c.clan_id=k.clan_id LEFT JOIN
citizen_status as j on c.status_id=j.status_id LEFT JOIN
land_proprietary as m on m.nik=c.nik LEFT JOIN land as l ON
l.nop=m.nop LEFT JOIN owner_state as d ON d.id_state=l.id_state
where MD5(c.clan_id::teks)='$current_selected_clan' AND
c.status_id='$current_selected_owner' AND
l.id_state='$current_selected_owner_land' AND
l.high_property='$current_selected_status'";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
$i=array();
while ($data = pg_fetch_assoc($Query)){
$data_a[$data['id']][] = $data;
}
return $data_a;
}}
public function
load_building_all4($current_selected_clan,$current_selected_owner,$
current_selected_owner_land){
$sql="SELECT b.*,f.family_no,c.name  FROM building b LEFT JOIN
building_family f ON b.id=f.id_building LEFT JOIN citizen c ON
c.nik=b.nik LEFT JOIN clans as k on c.clan_id=k.clan_id LEFT JOIN
citizen_status as j on c.status_id=j.status_id LEFT JOIN
land_proprietary as m on m.nik=c.nik LEFT JOIN land as l ON
l.nop=m.nop LEFT JOIN owner_state as d ON d.id_state=l.id_state
where MD5(c.clan_id::teks)='$current_selected_clan' AND
```

```
c.status_id='$current_selected_owner' AND
l.id_state='$current_selected_owner_land'";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
$i=array();
while ($data = pg_fetch_assoc($Query)){
$data_a[$data['id']][] = $data;
}
return $data_a;
}}
// Menampilkan List Bangunan Berupa Peta
// File Proses.php
else if($do == 'g_map_building_clan'){
header('Content-Type: application/json');
$current_selected_clan=$_GET['current_selected_clan'];
$current_selected_owner=$_GET['current_selected_owner'];
$current_selected_owner_land =$_GET['current_selected_owner_land'];
$current_selected_status=$_GET['current_selected_status'];
if($current_selected_clan == 'all' && $current_selected_owner ==
'all' && $current_selected_owner_land == 'all' &&
$current_selected_status == 'all'){
$m = new building();
$data = $m->load_building_map();
echo json_encode($data);
}
else if($current_selected_clan == 'all' && $current_selected_owner
!= 'all' && $current_selected_owner_land != 'all' &&
$current_selected_status != 'all'){
$m = new building();
$data = $m->load_map_building_all1($current_selected_owner,
$current_selected_owner_land,$current_selected_status );
echo json_encode($data);}
else if($current_selected_clan == 'all' && $current_selected_owner
!= 'all' && $current_selected_owner_land == 'all' &&
$current_selected_status == 'all'){
$m = new building();
data = $m->load_map_building_all7($current_selected_owner );
echo json_encode($data);
}
else if($current_selected_clan == 'all' && $current_selected_owner
== 'all' && $current_selected_owner_land != 'all' &&
$current_selected_status == 'all'){
$m = new building();
$data = $m->load_map_building_all3($current_selected_owner_land );
echo json_encode($data);
}
else if($current_selected_clan == 'all' && $current_selected_owner
== 'all' && $current_selected_owner_land == 'all' &&
$current_selected_status != 'all'){
$m = new building();
$data = $m->load_map_building_all4($current_selected_status);
echo json_encode($data);
}
```

c. Contoh Luaran Program

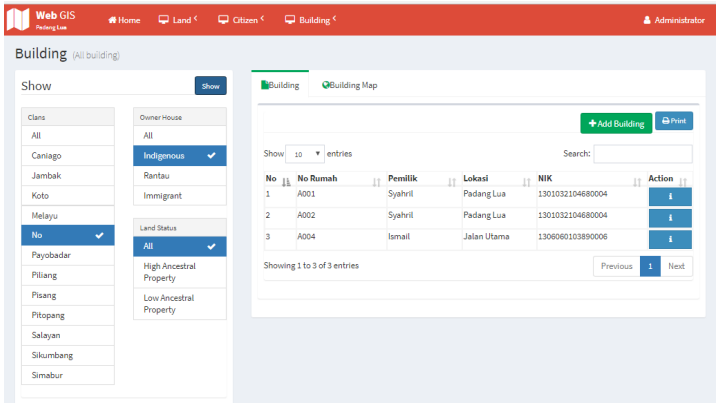
```
else if($current_selected_clan == 'all' && $current_selected_owner
== 'all' && $current_selected_owner_land != 'all' &&
$current_selected_status != 'all'){
$m = new building();
$data = $m-
>load_map_building_all5($current_selected_owner_land,$current_selec
ted_status);
echo json_encode($data);
}
else if(strlen($current_selected_clan) == 32 &&
$current_selected_owner == 'all' && $current_selected_owner_land ==
'all' && $current_selected_status == 'all'){
$m = new building();
$data = $m->load_map_building_all10($current_selected_clan);
echo json_encode($data);
}
else if(strlen($current_selected_clan) == 32 &&
$current_selected_owner != 'all' && $current_selected_owner_land ==
'all' && $current_selected_status == 'all'){
$m = new building();
$data = $m-
>load_map_building_all11($current_selected_clan,$current_selected_o
wner);
echo json_encode($data);
}
else if(strlen($current_selected_clan) == 32 &&
$current_selected_owner == 'all' && $current_selected_owner_land !=
'all' && $current_selected_status == 'all'){
$m = new building();
$data = $m-
>load_map_building_all12($current_selected_clan,$current_selected_o
wner_land);
echo json_encode($data);
}
else if(strlen($current_selected_clan) == 32 &&
$current_selected_owner == 'all' && $current_selected_owner_land ==
'all' && $current_selected_status != 'all'){
$m = new building();
$data = $m-
>load_map_building_all13($current_selected_clan,$current_selected_s
tatus);
echo json_encode($data);
}
else if(strlen($current_selected_clan) == 32 &&
$current_selected_owner == 'all' && $current_selected_owner_land !=
'all' && $current_selected_status != 'all'){
$m = new building();
$data = $m-
>load_map_building_all14($current_selected_clan,$current_selected_o
wner_land,$current_selected_status);
echo json_encode($data);
}
else if(strlen($current_selected_clan) == 32 &&
$current_selected_owner != 'all' && $current_selected_owner_land !=
'all' && $current_selected_status != 'all'){
```

```
$m = new building();
$data = $m-
>load_map_building_all15($current_selected_clan,$current_selected_o
wner,$current_selected_owner_land,$current_selected_status);
echo json_encode($data);
}
else{
echo 0;
}
}
-----------------------------------------------------------------
// file Class.php
public function load_building_map(){
$sql="SELECT ST_AsGeoJSON(geom) As geometry,
ST_Y(ST_CENTROID(geom)) AS lat, ST_X(ST_CENTROID(geom)) AS lng,
b.*,f.family_no,c.name ,case when  MD5(c.clan_id::teks)= 'all' then
0 else 1 end as a FROM building b LEFT JOIN building_family f ON
b.id=f.id_building LEFT JOIN citizen c ON c.family_no=f.family_no
LEFT JOIN citizen_status AS cs ON c.status_id = cs.status_id";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;
}
return $data_a;
}}
public function load_map_building_all1($current_selected_owner,
$current_selected_owner_land,$current_selected_status){
$sql="SELECT ST_AsGeoJSON(b.geom) As geometry,
ST_Y(ST_CENTROID(b.geom)) AS lat, ST_X(ST_CENTROID(b.geom)) AS lng,
b.*,f.family_no,c.name  ,case when
c.status_id='$current_selected_owner' AND l.id_state='
$current_selected_owner_land' AND
l.high_property='$current_selected_status'  then 1 else 0 end as a
FROM building b LEFT JOIN building_family f ON b.id=f.id_building
LEFT JOIN citizen c ON c.nik=b.nik LEFT JOIN clans as k on
c.clan_id=k.clan_id LEFT JOIN citizen_status as j on
c.status_id=j.status_id LEFT JOIN land_proprietary as m on
m.nik=c.nik LEFT JOIN land as l ON l.nop=m.nop LEFT JOIN
owner_state as d ON d.id_state=l.id_state";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;
}
return $data_a;
}}
public function
load_map_building_all15($current_selected_clan,$current_selected_ow
ner,$current_selected_owner_land,$current_selected_status){
$sql="SELECT ST_AsGeoJSON(b.geom) As geometry,
ST_Y(ST_CENTROID(b.geom)) AS lat, ST_X(ST_CENTROID(b.geom)) AS lng,
b.*,f.family_no,c.name,case when
MD5(c.clan_id::teks)='$current_selected_clan' AND
l.id_state='$current_selected_owner_land' AND
```

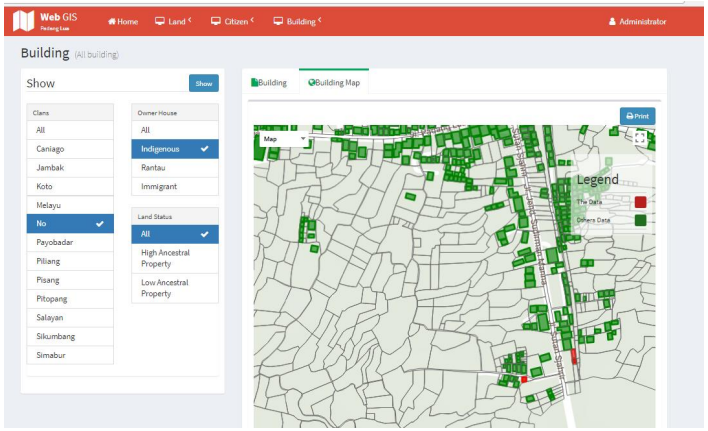
```
l.high_property='$current_selected_status'  then 1 else 0 end as a
FROM building b LEFT JOIN building_family f ON b.id=f.id_building
LEFT JOIN citizen c ON c.nik=b.nik LEFT JOIN clans as k on
c.clan_id=k.clan_id LEFT JOIN citizen_status as j on
c.status_id=j.status_id LEFT JOIN land_proprietary as m on
m.nik=c.nik LEFT JOIN land as l ON l.nop=m.nop LEFT JOIN
owner_state as d ON d.id_state=l.id_state";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;
}
return $data_a;
}}
public function
load_map_building_all14($current_selected_clan,$current_selected_ow
ner_land,$current_selected_status){
$sql="SELECT ST_AsGeoJSON(b.geom) As geometry,
ST_Y(ST_CENTROID(b.geom)) AS lat, ST_X(ST_CENTROID(b.geom)) AS lng,
b.*,f.family_no,c.name,case when
MD5(c.clan_id::teks)='$current_selected_clan' AND
c.status_id='$current_selected_owner' AND
l.id_state='$current_selected_owner_land' AND
l.high_property='$current_selected_status'  then 1 else 0 end as a
FROM building b LEFT JOIN building_family f ON b.id=f.id_building
LEFT JOIN citizen c ON c.nik=b.nik LEFT JOIN clans as k on
c.clan_id=k.clan_id LEFT JOIN citizen_status as j on
c.status_id=j.status_id LEFT JOIN land_proprietary as m on
m.nik=c.nik LEFT JOIN land as l ON l.nop=m.nop LEFT JOIN
owner_state as d ON d.id_state=l.id_state";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;
}
return $data_a;
}
}
public function
load_map_building_all13($current_selected_clan,$current_selected_st
atus){
$sql="SELECT ST_AsGeoJSON(b.geom) As geometry,
ST_Y(ST_CENTROID(b.geom)) AS lat, ST_X(ST_CENTROID(b.geom)) AS lng,
b.*,f.family_no,c.name,case when
MD5(c.clan_id::teks)='$current_selected_clan' AND
l.high_property='$current_selected_status'  then 1 else 0 end as a
FROM building b LEFT JOIN building_family f ON b.id=f.id_building
LEFT JOIN citizen c ON c.nik=b.nik LEFT JOIN clans as k on
c.clan_id=k.clan_id LEFT JOIN citizen_status as j on
c.status_id=j.status_id LEFT JOIN land_proprietary as m on
m.nik=c.nik LEFT JOIN land as l ON l.nop=m.nop LEFT JOIN
owner_state as d ON d.id_state=l.id_state";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
```

```
$data_a[] = $data;
}
return $data_a;
}}
public function
load_map_building_all12($current_selected_clan,$current_selected_ow
ner_land){
$sql="SELECT ST_AsGeoJSON(b.geom) As geometry,
ST_Y(ST_CENTROID(b.geom)) AS lat, ST_X(ST_CENTROID(b.geom)) AS lng,
b.*,f.family_no,c.name,case when
MD5(c.clan_id::teks)='$current_selected_clan' AND l.id_state=
'$current_selected_owner_land'  then 1 else 0 end as a   FROM
building b LEFT JOIN building_family f ON b.id=f.id_building LEFT
JOIN citizen c ON c.nik=b.nik LEFT JOIN clans as k on
c.clan_id=k.clan_id LEFT JOIN citizen_status as j on
c.status_id=j.status_id LEFT JOIN land_proprietary as m on
m.nik=c.nik LEFT JOIN land as l ON l.nop=m.nop LEFT JOIN
owner_state as d ON d.id_state=l.id_state";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;
}
return $data_a;
}}
public function
load_map_building_all11($current_selected_clan,$current_selected_ow
ner){
$sql="SELECT ST_AsGeoJSON(b.geom) As geometry,
ST_Y(ST_CENTROID(b.geom)) AS lat, ST_X(ST_CENTROID(b.geom)) AS lng,
b.*,f.family_no,c.name  ,case when
MD5(c.clan_id::teks)='$current_selected_clan' AND
c.status_id='$current_selected_owner' then 1 else 0 end as a   FROM
building b LEFT JOIN building_family f ON b.id=f.id_building LEFT
JOIN citizen c ON c.nik=b.nik LEFT JOIN clans as k on
c.clan_id=k.clan_id LEFT JOIN citizen_status as j on
c.status_id=j.status_id LEFT JOIN land_proprietary as m on
m.nik=c.nik LEFT JOIN land as l ON l.nop=m.nop LEFT JOIN
owner_state as d ON d.id_state=l.id_state";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;
}
return $data_a;
}}
public function load_map_building_all10($current_selected_clan){
$sql="SELECT ST_AsGeoJSON(b.geom) As geometry,
ST_Y(ST_CENTROID(b.geom)) AS lat, ST_X(ST_CENTROID(b.geom)) AS lng,
b.*,f.family_no,c.name  ,case when
MD5(c.clan_id::teks)='$current_selected_clan' then 1 else 0 end as
a   FROM building b LEFT JOIN building_family f ON
b.id=f.id_building LEFT JOIN citizen c ON c.nik=b.nik LEFT JOIN
clans as k on c.clan_id=k.clan_id LEFT JOIN citizen_status as j on
c.status_id=j.status_id LEFT JOIN land_proprietary as m on
m.nik=c.nik LEFT JOIN land as l ON l.nop=m.nop LEFT JOIN
owner_state as d ON d.id_state=l.id_state";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;
}
return $data_a;
}}
public function
load_map_building_all5($current_selected_owner_land,$current_select
ed_status){
$sql="SELECT ST_AsGeoJSON(b.geom) As geometry,
ST_Y(ST_CENTROID(b.geom)) AS lat, ST_X(ST_CENTROID(b.geom)) AS lng,
b.*,f.family_no,c.name  ,case when
l.id_state='$current_selected_owner_land' AND
l.high_property='$current_selected_status'  then 1 else 0 end as a
FROM building b LEFT JOIN building_family f ON b.id=f.id_building
LEFT JOIN citizen c ON c.nik=b.nik LEFT JOIN clans as k on
c.clan_id=k.clan_id LEFT JOIN citizen_status as j on
c.status_id=j.status_id LEFT JOIN land_proprietary as m on
m.nik=c.nik LEFT JOIN land as l ON l.nop=m.nop LEFT JOIN
owner_state as d ON d.id_state=l.id_state";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;
}
return $data_a;
}}
public function load_map_building_all4($current_selected_status){
$sql="SELECT ST_AsGeoJSON(b.geom) As geometry,
ST_Y(ST_CENTROID(b.geom)) AS lat, ST_X(ST_CENTROID(b.geom)) AS lng,
b.*,f.family_no,c.name  ,case when
l.high_property='$current_selected_status'  then 1 else 0 end as a
FROM building b LEFT JOIN building_family f ON b.id=f.id_building
LEFT JOIN citizen c ON c.nik=b.nik LEFT JOIN clans as k on
c.clan_id=k.clan_id LEFT JOIN citizen_status as j on
c.status_id=j.status_id LEFT JOIN land_proprietary as m on
m.nik=c.nik LEFT JOIN land as l ON l.nop=m.nop LEFT JOIN
owner_state as d ON d.id_state=l.id_state";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;
}
return $data_a;
}}
public function
load_map_building_all3($current_selected_owner_land){
$sql="SELECT ST_AsGeoJSON(b.geom) As geometry,
ST_Y(ST_CENTROID(b.geom)) AS lat, ST_X(ST_CENTROID(b.geom)) AS lng,
b.*,f.family_no,c.name  ,case when l.id_state='
$current_selected_owner_land' then 1 else 0 end as a   FROM
building b LEFT JOIN building_family f ON b.id=f.id_building LEFT
```

```
JOIN citizen c ON c.nik=b.nik LEFT JOIN clans as k on
c.clan_id=k.clan_id LEFT JOIN citizen_status as j on
c.status_id=j.status_id LEFT JOIN land_proprietary as m on
m.nik=c.nik LEFT JOIN land as l ON l.nop=m.nop LEFT JOIN
owner_state as d ON d.id_state=l.id_state";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;
}
return $data_a;
}}
public function load_map_building_all7($current_selected_owner){
$sql="SELECT ST_AsGeoJSON(geom) As geometry,
ST_Y(ST_CENTROID(geom)) AS lat, ST_X(ST_CENTROID(b.geom)) AS lng,
b.*,f.family_no,c.name  ,case when
c.status_id='$current_selected_owner'  then 1 else 0 end as a
FROM building b LEFT JOIN building_family f ON b.id=f.id_building
LEFT JOIN citizen c ON c.nik=b.nik LEFT JOIN clans as k on
c.clan_id=k.clan_id LEFT JOIN citizen_status as j on
c.status_id=j.status_id";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;
}
return $data_a;
}}
```



Gambar 13. Hasil Menampilkan *List* Bangunan Berupa Informasi Teks



Gambar 14. Hasil *List* Bangunan Berupa Peta

6. Mengelola Data Building History

a. Use Case Skenario

Tabel 11. Skenario Menampilkan Data Building History

Use Case Name	*Menampilkan data* Building history
Participating Actor	*User*
Flow Of Events	1. *User* memilih menu *land*
	2. Sistem akan menampilkan sub menu
	3. *User* memilih menu *land history*
	4. Sistem akan menampilkan data *building history*
Entry condition	*User* telah login
Exit Condition	*User* dapat melihat informasi *building history*

Tabel 12. Prosedur Menginputkan Data Building History

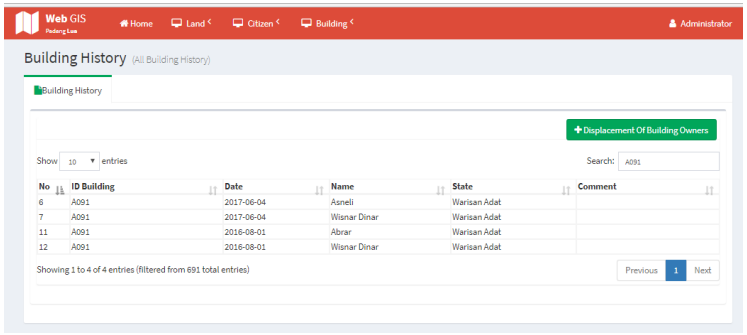
Use Case Name	*Menambah Data* Building history
Participating Actor	*User*
Flow Of Events	1. *User* memilih menu *building*
	2. Sistem akan menampilkan sub menu
	3. *User* memilih menu *building history*
	4. Sistem akan menampilkan menu *building history*
	5. *User* mengklik *buttton displacement of building owner*

	6. Sistem akan form menambah data *building history*
	7. *User* mengisi *form* dan memilih *button save*
	8. Sistem akan menyimpan perubahan dalam *database*
Entry condition	*User* telah login
Exit Condition	Sistem menyimpan perubahan data ke *database* dan membuka kembali halaman *building history*

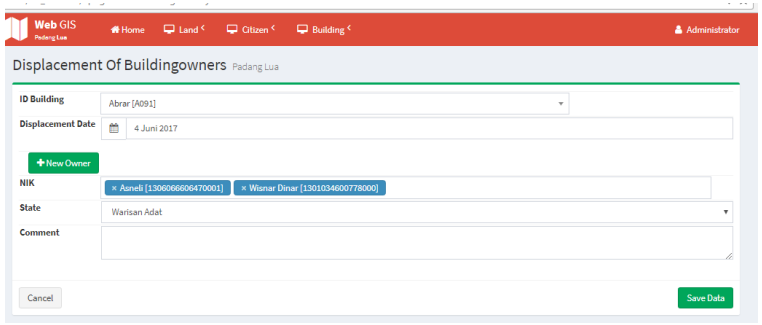
a. Kode Program

```
// Menampilkan Data Building History
//method untuk query menampilkan data Building  history
public function load_building_owner(){
$sql="SELECT  b.no_house, p.name, bo.id_building
FROM building as b LEFT JOIN building_owner as bo on
b.id=bo.id_building LEFT JOIN pemilik as p ON p.nik=
bo.pemilik_id order by name asc ";
$query = pg_query($sql);
if(pg_num_rows($query) > 0){
while ($data = pg_fetch_assoc($query)){
$data_a[] = $data;
}
return $data_a;
}}
// Menginputkan Data  Building History

//method untuk query menginputkan data history land
public function insert_building_history($data){
$sqlICN="INSERT INTO building_owner(pemilik_id, tgl, state,
coment, id_building) VALUES ";
$sqlICN.="(".$data['pemilik_id']."','".$data['tgl']."','".$da
ta['state']."','".$data['coment']."','".$data['id_building']."')"
;
$sql=$sqlICN.';';
$query = pg_query($sql);
if($query){
return 1;
}
else{
return 0;
}}
```

b. Contoh Luaran Program



Gambar 15. Hasil Menampilkan Data Building History



Gambar 16. Hasil Menginputkan Data BuildingHistory

7. Pencarian Kepemilikan Tanah Berdasarkan Nomor Penduduk

a. Use Case Skenario

Tabel 13. Skenario Pencarian Data Kepemilikan Tanah Berdasarkan Nomor Penduduk Berupa Informasi Teks

Use case name	Pencarian Data Kepemilikan Tanah Berdasarkan Nomor Penduduk Berupa Informasi Teks
Participating Actor	*User*
Flow Of Events	1. *User* memilih menu *citizen*
	2. Sistem akan menampilkan sub menu

	3. *User* memilih menu *land properietary search*
	4. *Sistem* akan menampilkan menu *land properietary search*
	5. *User* memasukan nomor penduduk
	6. Sistem akan menampilkan informasi teks mengenai data penduduk
Entry condition	*User* telah login
Exit Condition	*User* dapat melihat informasi teks mengenai data penduduk yang diinginkan
Alternative Course	1. *User* memilih menu *land*
	2. Sistem akan menampilkan sub menu
	3. *User* memilih menu *land search*
	4. *Sistem* akan menampilkan menu *land properietary search*
	5. *User* memasukan nomor penduduk
	6. Sistem akan menampilkan "*Data Not Found'*

Tabel 14. Skenario Pencarian Data Kepemilikan Tanah Berdasarkan Nomor Penduduk Berupa Peta

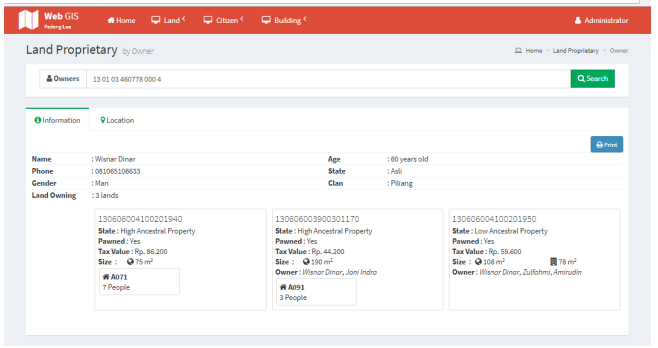
Use case name	Pencarian Data Kepemilikan Tanah Berdasarkan Nomor Penduduk Berupa Peta
Participating Actor	*User*
Flow Of Events	a. *User* memilih menu *citizen*
	b. Sistem akan menampilkan sub menu
	c. *User* memilih menu *land properietary search*
	d. Sistem akan menampilkan menu *land properietary search*
	e. *User* memasukan nomor penduduk
	f. Sistem akan menampilkan data nomor penduduk
	g. *User* memilih tab *map*
	7. Sistem akan menampilkan lokasi berdasarkan nomor penduduk yang diinputkan
Entry condition	*User* telah login
Exit Condition	*User* dapat melihat lokasi berdasarkan nomor penduduk

b. Kode Program

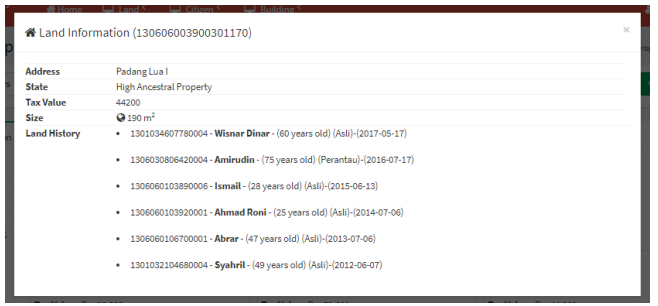
```
// Pencarian Data Penduduk Berdasarkan Nomor Penduduk Berupa Informasi Teks Untuk Kepemilikian Tanah

// Method untuk memanggil data dari database menggunakan
// bahasa pemrograman php

public function find_owner($id){
// query pencarian data penduduk berdasarkan nomor penduduk
$sql = "SELECT ci.nik, ci.name, ci.phone, ci.gender,
EXTRACT(YEAR FROM AGE(ci.born_date)) AS age, cs.name AS
status, cl.name AS clan FROM citizen AS ci LEFT JOIN
citizen_status AS cs ON ci.status_id = cs.status_id LEFT JOIN
clans AS cl on ci.clan_id = cl.clan_id WHERE ci.nik = '$id'";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
//generate informasi data penduduk;
$result = pg_fetch_assoc($Query)
$data['nik'] = $result['nik'];
$data['name'] = $result['name'];
$data['phone'] = $result['phone'];
$data['gender'] = $result['gender'];
$data['age'] = $result['age'];
$data['status'] = $result['status'];
$data['clan'] = $result['clan'];
//memilih tanah yang berdasarkan data penduduk yang
diinputkan;
$sql = "SELECT la.nop, la.high_property, la.tax_value,
la.pawning, la.land_size, la.building_size FROM
land_proprietary As lp LEFT JOIN land AS la ON lp.nop = la.nop
WHERE lp.nik = '".$result['nik']."'";
$Query_object = pg_Query($sql);
$data_object = null;
if(pg_num_rows($Query_object)>0){
$data_object = array();
while ($result_sub = pg_fetch_assoc($Query_object)) {
//memilih pemilik dari tanah;
$sql = "SELECT sci.nik, sci.name AS name, EXTRACT(YEAR FROM
AGE(sci.born_date)) AS age FROM land_proprietary As slp LEFT
```

c. Contoh luaran program

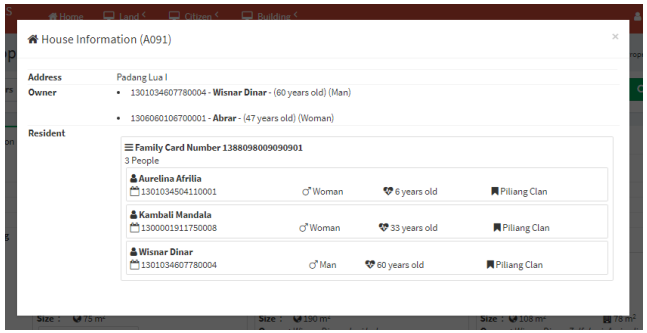
```
JOIN citizen AS sci ON slp.nik = sci.nik WHERE slp.nop =
'".$result_sub['nop']."'";
$Query_owner = pg_Query($sql);
$data_owner = null;
if(pg_num_rows($Query_owner)){
$data_owner = array();
while ($result_sub_re = pg_fetch_assoc($Query_owner)) {
$data_owner[] = $result_sub_re;
}}
$data_object[] = array(
'nop' => $result_sub['nop'],
'high_property' => $result_sub['high_property'],
'tax_value' => $result_sub['tax_value'],
'pawning' => $result_sub['pawning'],
'land_size' => $result_sub['land_size'],
'building_size' => $result_sub['building_size'],
'owner' => $data_owner,
);
}}
     $data['object'] = $data_object;
     return $data;
     }
     else{
     return 0;
     }}

// Pencarian Data Penduduk Berdasarkan Nomor Penduduk Untuk Kepemilikan Tanah
// Berupa Peta
//method mendapatkan layer dari object bangunan berdasarkan
nomor penduduk
public function find_object_tax($id){
$sql = "SELECT la.nop, la.high_property, la.pawning,
la.tax_value, la.land_size, la.building_size, (SELECT
array_to_json(array_agg(f)) FROM (SELECT ci.nik, ci.name AS
name, cs.name AS status, EXTRACT(YEAR FROM AGE(ci.born_date))
AS age FROM tax_subject As ts LEFT JOIN citizen AS ci ON
ts.nik = ci.nik LEFT JOIN citizen_status AS cs ON ci.status_id
= cs.status_id WHERE ts.nop = la.nop) AS f) AS subject,
jo.name AS address FROM land AS la, jorong AS jo WHERE
ST_CONTAINS(jo.geom, ST_CENTROID(la.geom)) AND la.nop =
'$id'";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
$result = pg_fetch_assoc($Query);
$data['nop'] = $result['nop'];
$data['high_property'] = $result['high_property'];
$data['pawning'] = $result['pawning'];
$data['tax_value'] = $result['tax_value'];
$data['land_size'] = $result['land_size'];
$data['building_size'] = $result['building_size'];
$data['subject'] = json_decode($result['subject']);
$data['address'] = $result['address'];
return $data;
}
else{
return 0;
}}
```



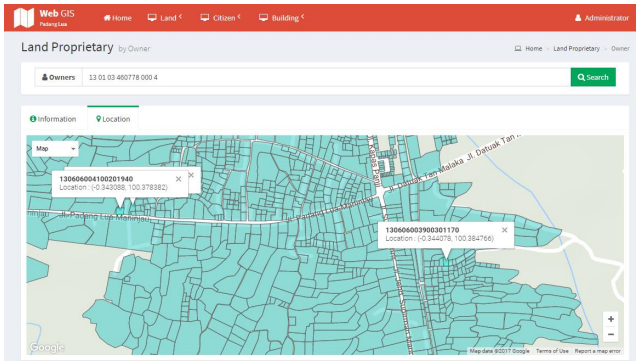
Gambar 17. Hasil Data Normal Pencarian Data Kepemilikan Tanah Berdasarkan Nomor Penduduk Berupa Informasi Teks



Gambar 18. Land Information



Gambar 19. *House Information*



Gambar 20. Hasil Pencarian Data Kepemilikan Tanah Berdasarkan Nomor Penduduk Berupa Peta

8. Pencarian Data Kepemilikan Bangunan Berdasarkan Nomor Penduduk

a. Use Case Skenario

Tabel 15. Skenario Pencarian Data Kepemilikan Bangunan Berdasarkan Nomor Penduduk Berupa Informasi Teks

Use case name	Pencarian Data Kepemilikan Bangunan Berdasarkan Nomor Penduduk Berupa Informasi Teks
Participating Actor	*User*
Flow Of Events	1. *User* memilih menu *citizen*
	2. Sistem akan menampilkan sub menu
	3. *User* memilih sub menu *building owner search*
	4. *Sistem* akan menampilkan sub menu *building owner search*
	5. *User* memasukan nomor penduduk
	6. Sistem akan menampilkan informasi teks mengenai nomor penduduk
Entry condition	*User* telah login
Exit Condition	*User* dapat melihat informasi teks mengenai data pendudukyang diinginkan
Alternative Course	5. *User* memilih menu *citizen*
	6. Sistem akan menampilkan sub menu
	7. *User* memilih sub menu *building owner search*
	8. *Sistem* akan menampilkan sub menu *building owner search*
	7. *User* memasukan nomor penduduk
	8. Sistem akan menampilkan "*Data Not Found'*

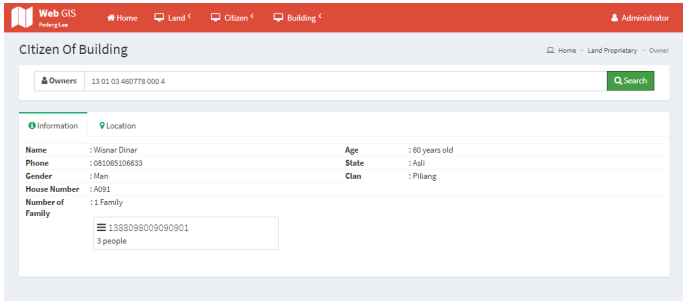
Tabel 16. Skenario Pencarian Data Kepemilikan Bangunan Berdasarkan Nomor Penduduk Berupa Peta

Use case name	Pencarian Data Kepemilikan Tanah Berdasarkan Nomor Penduduk Berupa Peta
Participating Actor	*User*
Flow Of Events	a. *User* memilih menu *citizen*
	b. Sistem akan menampilkan sub menu
	c. *User* memilih menu *building owner search*
	d. Sistem akan menampilkan menu *building owner search*
	e. *User* memasukan nomor penduduk
	f. Sistem akan menampilkan data nomor penduduk
	g. *User* memilih tab *map*
	8. Sistem akan menampilkan lokasi berdasarkan nomor penduduk
Entry condition	*User* telah login
Exit Condition	*User* dapat melihat lokasi berdasarkan nomor penduduk

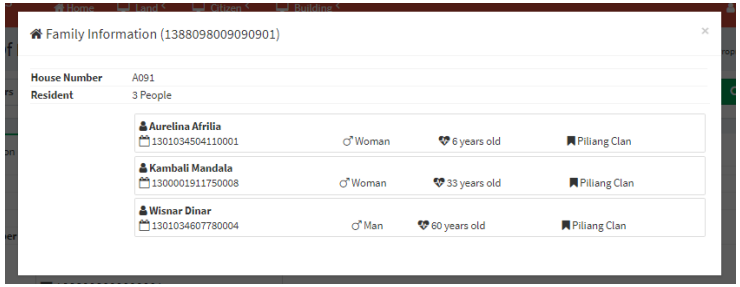
b. Kode Program

```
// Pencarian Data Penduduk Berdasarkan Nomor Penduduk Berupa Informasi
// Teks Untuk Kepemilikian Bangunan
//Method untuk memanggil data dari database menggunakan bahasa
//pemrograman php
public function find_owner1($id){
//query pencarian data penduduk
$sql = "SELECT ci.nik, ci.name, ci.phone, ci.gender,
EXTRACT(YEAR FROM AGE(ci.born_date)) AS age, cs.name AS status,
cl.name AS clan, bd.id, ci.family_no FROM building as bd left
join building_family as bf ON bd.id=bf.id_building left join
citizen AS ci on ci.family_no=bf.family_no LEFT JOIN
citizen_status AS cs ON ci.status_id = cs.status_id LEFT JOIN
clans AS cl on ci.clan_id = cl.clan_id WHERE ci.nik ='$id'";
$query = pg_query($sql);
if(pg_num_rows($query) > 0){
//generate informasi data penduduk;
$result = pg_fetch_assoc($query);
$data['nik'] = $result['nik'];
$data['name'] = $result['name'];
$data['phone'] = $result['phone'];
$data['gender'] = $result['gender'];
$data['age'] = $result['age'];
$data['status'] = $result['status'];
$data['clan'] = $result['clan'];
$data['id'] = $result['id'];
$data['family_no'] = $result['family_no'];
//memilih tanah yang dimiliki oleh data penduduk;
$sql = "SELECT la.nop, la.high_property, la.tax_value,
la.pawning, la.land_size, la.building_size FROM
land_proprietary As lp LEFT JOIN land AS la ON lp.nop = la.nop
WHERE lp.nik = '".$result['nik']."'";
$query_object = pg_query($sql);
$data_object = null;
if(pg_num_rows($query_object)>0){
$data_object = array();
while ($result_sub = pg_fetch_assoc($query_object)) {
//memilih pemilik tanah;
$sql = "SELECT sci.nik, sci.name AS name, EXTRACT(YEAR FROM
AGE(sci.born_date)) AS age FROM land_proprietary As slp LEFT
JOIN citizen AS sci ON slp.nik = sci.nik WHERE slp.nop =
'".$result_sub['nop']."'";
$query_owner = pg_query($sql);
$data_owner = null;
if(pg_num_rows($query_owner)){
$data_owner = array();
while ($result_sub_re = pg_fetch_assoc($query_owner)) {
$data_owner[] = $result_sub_re;
}}
$data_object[] = array(
'nop' => $result_sub['nop'],
'high_property' => $result_sub['high_property'],
tax_value' => $result_sub['tax_value'],
'pawning' => $result_sub['pawning'],
'land_size' => $result_sub['land_size'],
'building_size' => $result_sub['building_size'],
'owner' => $data_owner,
); } }
$data['object'] = $data_object;
//mencari informasi pemilik tanah
$sql = "SELECT ci.family_no, COUNT(ci.nik) AS resident
FROM citizen AS ci LEFT JOIN building_family as b ON
ci.family_no=b.family_no LEFT JOIN building as c ON
b.id_building=c.id  WHERE id = '".$result['id']."' GROUP BY
ci.family_no";
$query_fcn = pg_query($sql);
$data_fcn = null;
if(pg_num_rows($query_fcn)>0){
$data_fcn = array();
while ($result_sub_fcn = pg_fetch_assoc($query_fcn)) {
$data_fcn[] = $result_sub_fcn;
}}
$data['resident'] = $data_fcn;
return $data;
}
else{
return 0;
}}
// Pencarian Data Penduduk Berdasarkan Nomor Penduduk Untuk Kepemilikan
// Bangunan Berupa Peta
//method mendapatkan layer dari object bangunan berdasarkan
//nomor penduduk
public function object_layer_owner($owner_id){
$sql  = "SELECT ST_AsGeoJSON(la.geom) AS geometry, la.nop,
ST_Y(ST_CENTROID(la.geom)) AS lat, ST_X(ST_CENTROID(la.geom))
AS lng FROM land AS la LEFT JOIN land_proprietary AS lp ON
la.nop = lp.nop WHERE lp.nik = '$owner_id'";
$geojson = array(
'type'      => 'FeatureCollection',
'features'  => array()
);
$query = pg_query($sql);
if(pg_num_rows($query)==0) return 0;
while($edge=pg_fetch_assoc($query)){
$feature = array(
"type" => 'Feature',
'geometry' => json_decode($edge['geometry'], true),
'properties' => array(
'object' => $edge['nop'],
'center' => array(
'lat' => $edge['lat'],
'lng' => $edge['lng']
)));
array_push($geojson['features'], $feature);
}
return $geojson;
}
```

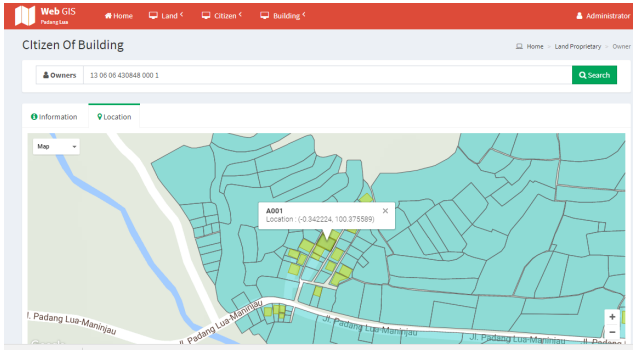
c. Contoh Luaran Program



Gambar 21. Hasil Data Normal Pencarian Data Kepemilikan Bangunan Berdasarkan Nomor Penduduk Berupa Informasi Teks



Gambar 22. Family Information



Gambar 23. Hasil Pencarian Data Kepemilikan Bangunan Berdasarkan Nomor Penduduk Berupa Peta

9. Pencarian Data Penduduk Berdasarkan Nomor Penduduk

a. Use Case Skenario

Tabel 17. Skenario Pencarian Data Penduduk Berdasarkan Nomor Penduduk Berupa Informasi Teks

Use Case Name	Pencarian Penduduk Berdasarkan Nomor Penduduk Berupa Informasi Teks
Participating Actor	*User*
Flow Of Events	1. *User* memilih menu *citizen*
	2. Sistem akan menampilkan submenu
	3. *User* memilih submenu *population search*
	4. Sistem akan menampilkan submenu *population search*
	5. *User* menginputkan salah satu nomor penduduk
	6. Sistem menampilkan data berdasarkan nomor yang diinputkan
Entry condition	*User* telah login
Exit Condition	Sistem Menampilkan data bedasarkan kriteria yang dipilih

Tabel 18. Skenario Pencarian Data Penduduk Berdasarkan Nomor penduduk Berupa Informasi Peta

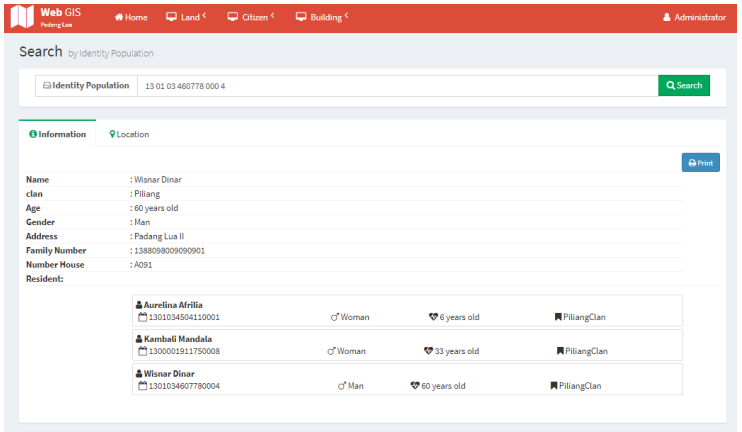
Use Case Name	Pencarian Penduduk Berdasarkan Nomor Penduduk Berupa Peta
Participating Actor	*User*
Flow Of Events	1. *User* memilih menu *citizen*
	2. Sistem akan menampilkan submenu
	3. *User* memilih submenu *population search*
	4. Sistem akan menampilkan submenu *population search*
	5. *User* menginputkan salah satu nomor penduduk

	6. Sistem menampilkan informasi berdasarkan nomor penduduk yang diinputkan
	7. *User* memilih tab *map*
	8. Sistem menampilkan halaman tab *map*
Entry condition	*User* telah login
Exit Condition	Sistem Menampilkan data bedasarkan kriteria yang dipilih

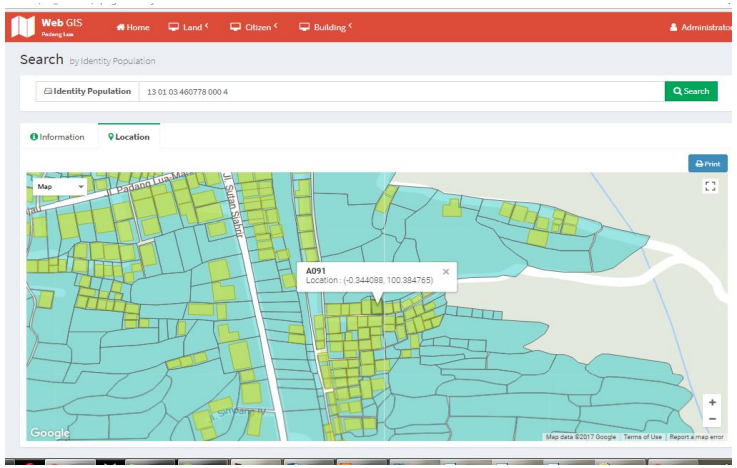
b. Kode Program

```
// Pencarian Data Penduduk Berdasarkan Nomor Penduduk Berupa
// Informasi Teks
//Method untuk memanggil data dari database menggunakan
//bahasa pemrograman php
public function find_family($family_id){
$sql = "SELECT re.id_building, COUNT(re.family_no) AS amount,
ci.family_no, ci.address, ci.name, cl.name as clan,
ci.gender, EXTRACT(YEAR FROM AGE(ci.born_date)) AS age FROM
building_family AS re LEFT JOIN citizen AS ci ON
re.family_no= ci.family_no left join clans as cl on
cl.clan_id=ci.clan_id WHERE  ci.nik ='".$family_id."' GROUP
BY re.id_building, ci.family_no,ci.address, ci.name, cl.name,
ci.gender, ci.born_date";
$query = pg_query($sql);
if(pg_num_rows($query) > 0){
$result = pg_fetch_assoc($query);
$data['family_no'] = $result['family_no'];
$data['id_building'] = $result['id_building'];
$data['amount'] = $result['amount'];
$data['addres'] = $result['address'];
$data['name'] = $result['name'];
$data['clan'] = $result['clan'];
$data['gender'] = $result['gender'];
$data['age'] = $result['age'];
//mencari daftar anggota
$sql = "SELECT ci.nik, ci.name, ci.gender, EXTRACT(YEAR FROM
AGE(ci.born_date)) AS age, cl.name AS clan FROM citizen AS ci
LEFT JOIN clans AS cl ON ci.clan_id=cl.clan_id WHERE
ci.family_no ='".$result['family_no']."'";
$query_people = pg_query($sql);
$data_people = null;
if(pg_num_rows($query_people)>0){
$data_people = array();
while ($result_sub = pg_fetch_assoc($query_people)) {
$data_people[] = $result_sub;
} }
$data['people'] = $data_people;
return $data;
}
else{
return 0;}}
// Pencarian Data Penduduk Berdasarkan Nomor Penduduk Berupa Peta
// method mendapatkan layer dari object bangunan berdasarkan
// nomor penduduk
public function family_layer($family_id){
$sql  = "SELECT ST_AsGeoJSON(ho.geom) As geometry, ho.id,
ST_Y(ST_CENTROID(ho.geom)) AS lat, ST_X(ST_CENTROID(ho.geom))
AS lng FROM building As ho left join building_family as bf on
bf.id_building=ho.id left join citizen as ci on
ci.family_no=bf.family_no WHERE ci.nik = '$family_id'";
$geojson = array(
'type'      => 'FeatureCollection',
'features'  => array()
);
$query = pg_query($sql);
if(pg_num_rows($query)==0) return 0;
while($edge=pg_fetch_assoc($query)){
$feature = array(
"type" => 'Feature',
'geometry' => json_decode($edge['geometry'], true),
'properties' => array(
'object' => $edge['id'],
'center' => array(
'lat' => $edge['lat'],
'lng' => $edge['lng']
)));
array_push($geojson['features'], $feature);
}
return $geojson;}
```

c. **Contoh luaran program**



Gambar 24. Hasil Data Normal Pencarian Data Penduduk Berdasarkan Nomor Penduduk Berupa *Informasi Teks*



Gambar 25. Hasil Pencarian Data Penduduk Berdasarkan Nomor Penduduk Berupa Peta

10. Mengelola Data Kelahiran

a. Use Case Skenario

Tabel 19. Prosedur Fungsional Menampilkan Peta Berdasarkan List Data Kelahiran

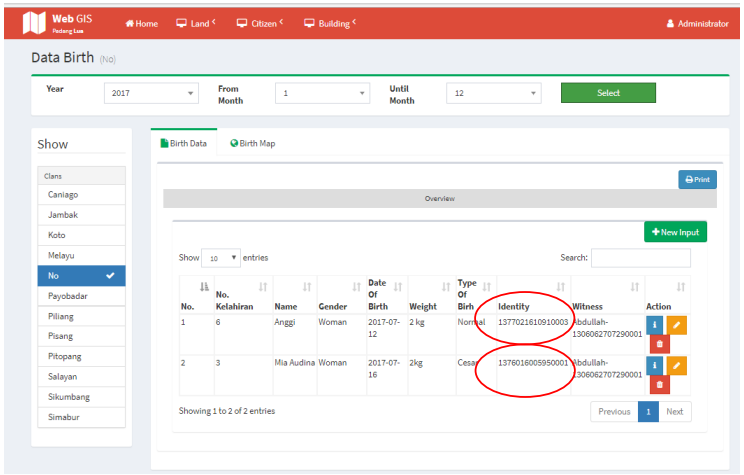
Use Case Name	Menampilkan Persebaran Peta Kelahiran
Participating Actor	*User*
Flow Of Events	1. *User* memilih menu *citizen*
	2. Sistem menampilkan submenu dari menu *citizen*
	3. *User* memilih menu data kelahiran
	4. Sistem menampilkan halaman data kelahiran
	5. *User* memilih tab *map*
	6. Sistem akan menampilkan halaman *map*
	7. *User* memilih kriteria yang telah disediakan dan mengklilk *button search*
	8. Sistem menampilkan persebaran peta berdasarkan kriteria yang di pilih sebelumnya

Entry condition	*User* telah *login*
Exit Condition	Sistem menampilkan persebaran peta

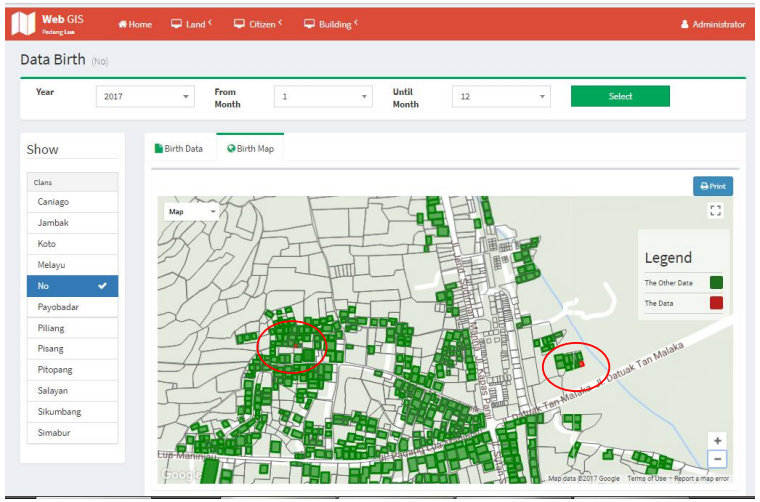
b. Kode program

```
// Menampilkan Peta Berdasarkan Kelahiran
// method untuk menampilkan layer berdasarkan data kelahiran
public function status_map_birth_clan($selected,$year,$m1,$m2){
$sql = "SELECT ST_AsGeoJSON(geom) As geometry,
ST_Y(ST_CENTROID(geom)) AS lat, ST_X(ST_CENTROID(geom)) AS lng,
building.id,birth_date, case when MD5(citizen.clan_id::teks) =
'$selected' AND EXTRACT (MONTH from out_come_date) >= '$m1' AND
EXTRACT (MONTH from out_come_date) <= '$m2' AND EXTRACT (year from
birth_data.birth_date)!='$year' or EXTRACT (year from
birth_data.birth_date) is NULL then 0 else 1 end as a FROM
birth_data right join citizen on
birth_data.family_no=citizen.family_no right join building on
building.family_no=citizen.family_no order by  birth_date";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;
}
return $data_a;
}else{
return 0;
}}
```

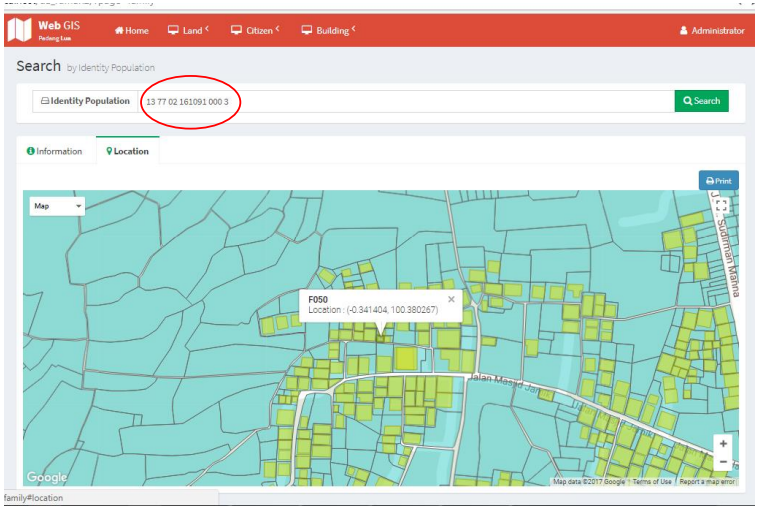
c. Contoh Luraran Program



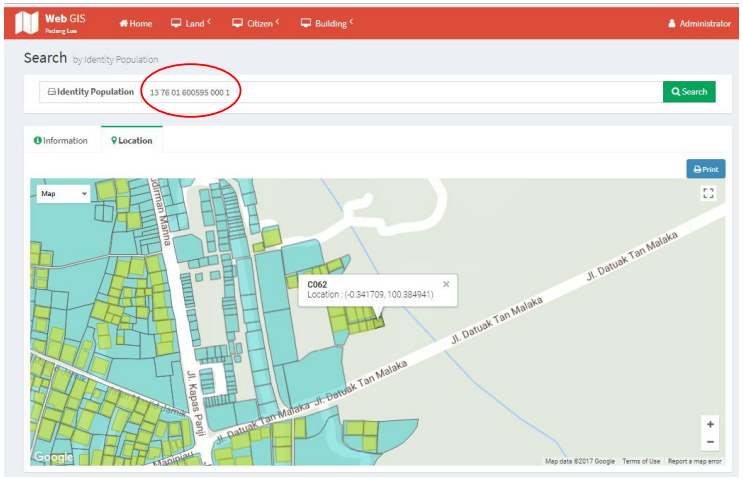
Gambar 26. Hasil Menampilkan *List* data Kelahiran



Gambar 27. Hasil Menampilkan Peta Berdasarkan *List* data Kelahiran



Gambar 28. Data Pada Pencarian Data Penduduk Berdasarkan Nomor Penduduk (1)



Gambar 29. Data Pada Pencarian Data Penduduk Berdasarkan Nomor Penduduk(2)

11. Mengelola Data Kematian

a. Use Case Skenario

Tabel 20. Skenario Menampilkan Peta Berdasarkan Data Kematian

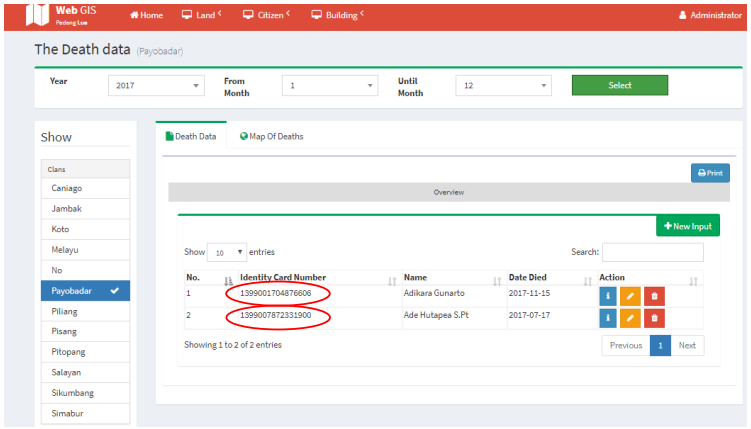
Use Case Name	Menampilkan Persebaran Peta Kematian
Participating Actor	*User*
Flow Of Events	1. *User* memilih menu *citizen*
	2. Sistem menampilkan submenu dari menu *citizen*
	3. *User* memilih menu data kematian
	4. Sistem menampilkan halaman data kematian
	5. *User* memilih tab *map*
	6. Sistem akan menampilkan halaman *map*
	7. *User* memilih kriteria yang telah disediakan dan mengklilk *button search*
	8. Sistem menampilkan persebaran peta berdasarkan kriteria yang di pilih sebelumnya
Entry condition	*User* telah login

Exit Condition	Sistem menampilkan persebaran peta

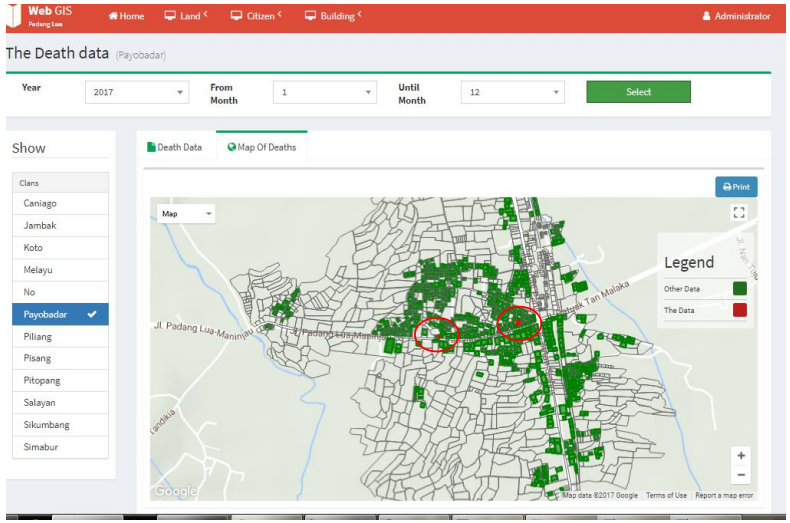
b. Kode Program

```
//  Menampilkan Peta Berdasarkan Kematian Penduduk
// method untuk menampilkan layer berdasarkan data kematian
public function
status_map_dead_clan($selected,$year,$m1,$m2){
$sql = "SELECT ST_AsGeoJSON(geom) As geometry,
ST_Y(ST_CENTROID(geom)) AS lat, ST_X(ST_CENTROID(geom)) AS
lng, building.id,death_date, case when
MD5(citizen.clan_id::teks) = '$selected' AND EXTRACT (MONTH
from mortality_data.death_date) >= '$m1' AND EXTRACT (MONTH
from mortality_data.death_date) <= '$m2' AND EXTRACT (year
from mortality_data.death_date) ='$year' AND EXTRACT (year
from mortality_data.death_date) is NOT NULL then 1 else 0 end
as a FROM mortality_data right join citizen on
mortality_data.nik=citizen.nik right join building on
building.nik=citizen.nik order by death_date ";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;
}
return $data_a;
}
else{
return 0;
}}
```

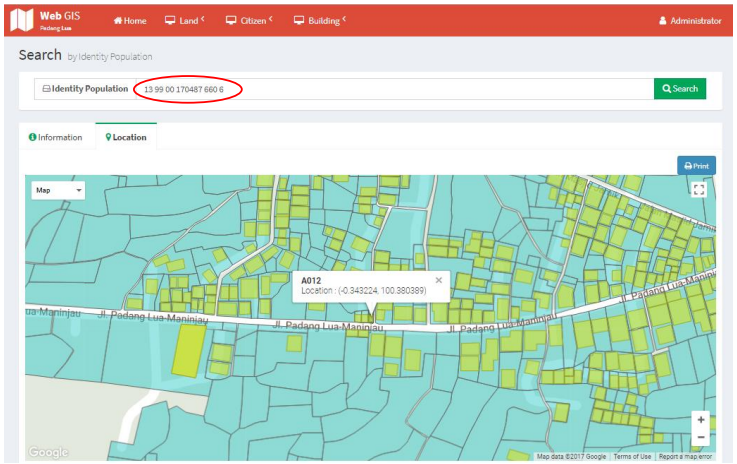
c. Contoh Luaran Program



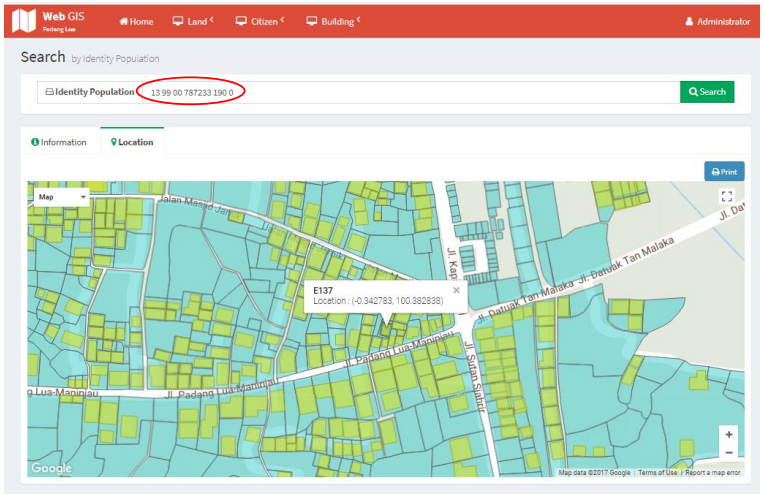
Gambar 6. Hasil Menampilkan Data Kematian



Gambar 7. Hasil Menampilkan Peta Berdasarkan Data Kematian



Gambar 8. Data Menggunakan Data Penduduk Berdasarkan Nomor Penduduk (1)



Gambar 9. Data Menggunakan Data Penduduk Berdasarkan Nomor Penduduk

12. Mengelola Perpindahan Penduduk

a. Use case skenario

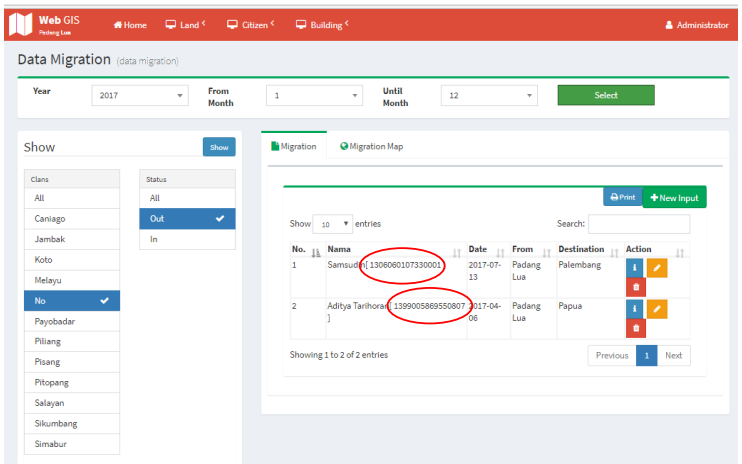
Tabel 21. Skenario Menampilkan Peta Berdasarkan Data Perpindahan Penduduk

Use Case Name	Menampilkan Persebaran Peta Perpindahan
Participating Actor	*User*
Flow Of Events	1. *User* memilih menu *citizen*
	2. Sistem menampilkan submenu dari menu *citizen*
	3. *User* memilih menu data perpindahan
	4. Sistem menampilkan halaman data perpindahan
	5. *User* memilih tab *map*
	6. Sistem akan menampilkan halaman *map*
	7. *User* memilih kriteria yang telah disediakan dan mengklilk *button search*
	8. Sistem menampilkan persebaran peta berdasarkan kriteria yang di pilih sebelumnya
Entry condition	*User* telah login
Exit Condition	Sistem menampilkan persebaran peta

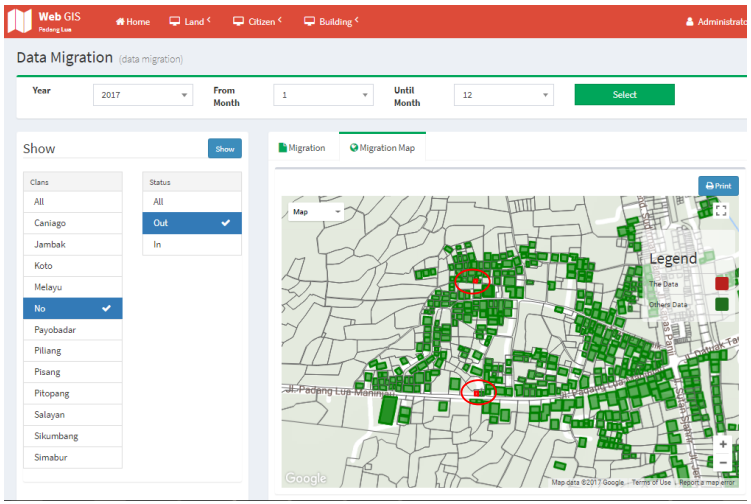
b. Kode Program

```
// Menampilkan Peta Berdasarkan Perpindahan
// method untuk menampilkan layer berdasarkan data perpindahan
public function
status_map_outcome_clan($selected,$year,$m1,$m2){
$sql = "SELECT ST_AsGeoJSON(geom) As geometry,
ST_Y(ST_CENTROID(geom)) AS lat, ST_X(ST_CENTROID(geom)) AS lng,
building.id,out_come_date, case when MD5(citizen.clan_id::teks)
= '$selected' AND EXTRACT (MONTH from out_come_date) >= '$m1'
AND EXTRACT (MONTH from out_come_date) <= '$m2' AND  EXTRACT
(year from out_come.out_come_date) = '$year' AND EXTRACT (year
from out_come.out_come_date)  is NOT NULL then 1 else 0 end as
a FROM out_come right join citizen on out_come.nik=citizen.nik
right join building on building.nik=citizen.nik  order by
out_come_date ";
$Query = pg_Query($sql);
if(pg_num_rows($Query) > 0){
while ($data = pg_fetch_assoc($Query)){
$data_a[] = $data;}
return $data_a;
}  else{
return 0;} }
```

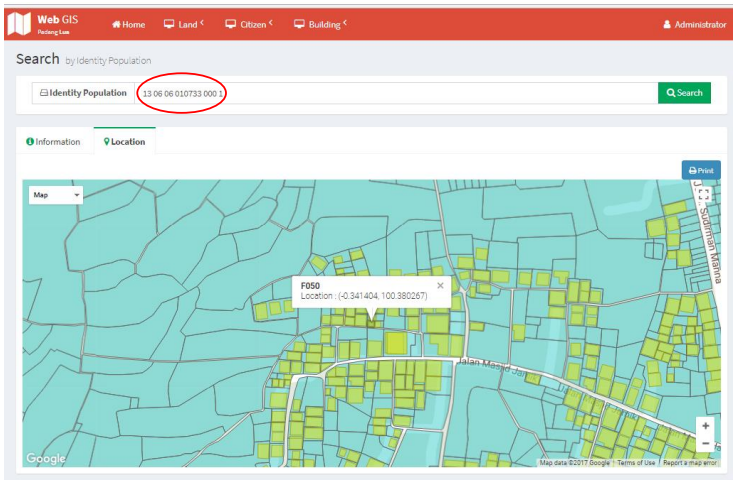
c. Contoh Luaran Program



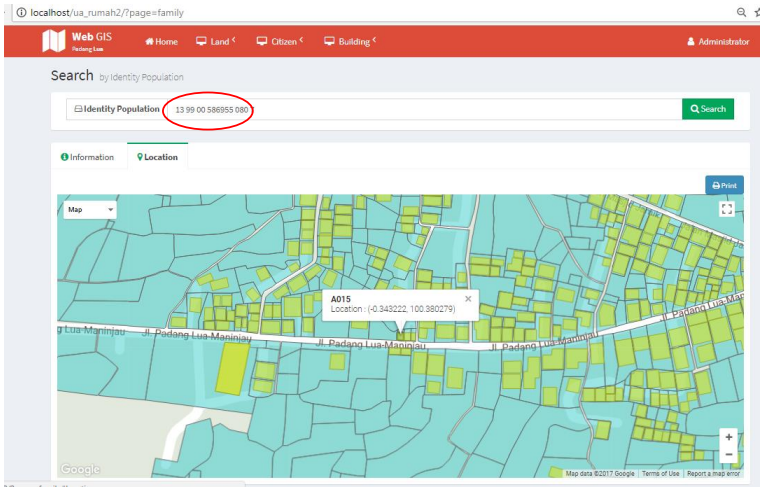
Gambar 10. Hasil Menampilkan Data Perpindahan Penduduk



Gambar 11. Hasil Menampilkan Peta Berdasarkan Data Perpindahan Penduduk



Gambar 12. Data Menggunakan Pencarian Data Penduduk Berdasarkan Nomor Penduduk (1)



Gambar 13. Data Menggunakan Pencarian Data Penduduk Berdasarkan Nomor Penduduk(2)